

北京航空航天大学出版社



王玮

研发主管。长期从事电子产品的软硬件研发工作。喜欢读书且常常参与实践，所以对很多事物有自己的独特见解。在EDN上的个人博客为《远去的村庄》。

EDN博客
http://blog.ednchina.com/herald/
《感悟设计》书友会
■http://group.ednchina.com/1638/
■http://bbs.21ic.com/club/bbs/list.asp?boardid=69

【博客藏经阁丛书】

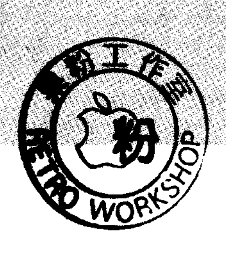
感悟设计



电子设计的经验与哲理



王玮 编著
【网名 大道至简】



北京航空航天大学出版社

内 容 简 介

本书主要从设计者的思维角度来展开叙述。但和别的讲设计的书籍显著的不同在于：本书不讲某个专题，而是注重描述在设计的整个过程中的思维方式和解决问题的方法。本书虽作为一本讲述电子设计的书籍，在其中穿插了很多非电子的案例和知识点。同时，也引用了一些大家都耳熟能详的哲学道理，并把这些道理作为文章的标题，用案例来陈述这些哲学道理的合理性。

本书适合于电子设计开发人员、管理人员和培训人员阅读参考，也可以作为学习电子设计的研究生和本科生的参考读物。

图书在版编目(CIP)数据

感悟设计：电子设计的经验与哲理/王玮编著. —北京：北京航空航天大学出版社，2009.5
ISBN 978-7-81124-573-8

Ⅰ. 感… Ⅱ. 王… Ⅲ. 电子电路—电路设计 Ⅳ. TN702

中国版本图书馆 CIP 数据核字(2009)第 047601 号

感悟设计
电子设计的经验与哲理
王　玮[网名　大道至简]　编著
责任编辑　陈　旭　敖惠珍
*
北京航空航天大学出版社出版发行
北京市海淀区学院路 37 号(100191)　发行部电话：010-82317024　传真：010-82328026
http://www.buaapress.com.cn　E-mail:emsbook@gmail.com
北京时代华都印刷有限公司印装　各地书店经销
*
开本：787 mm×960 mm　1/16　印张：15.5　字数：347 千字
2009 年 5 月第 1 版　2009 年 5 月第 1 次印刷　印数：5 000 册
ISBN 978-7-81124-573-8　　定价：32.00 元

序

一道原汁原味的“土家菜”

《感悟设计:电子设计的经验与哲理》是一本什么样的书？乍一看到这书名，相信读者您和匠人一样好奇。翻开这本书后，匠人找到了答案。这正是一本匠人希望看到的书，来自电子行业一线工程师的心血总结，类似于武功秘籍的那种。

在这本书中，作者把他工作十几年积累下来的经验汇总成册，做成一道原汁原味的“土家菜”端到读者面前。这里面没有僵化的或者人云亦云的教条主义，也没有云里雾里晕死你不偿命的理论公式，有的都是作者自己实实在在的工作经验和设计感悟。

当我们顺着作者的娓娓叙说，跟着他的行文方向前进，就仿佛亲身经历了他所经历过的那些实践一般。那种感觉就好像一个武功大师把毕生功力输入你的体内，打通你的经脉，让你豁然开朗，经验值暴涨。这可是打着灯笼难找的好事，呵呵，你就偷着乐吧！

而更重要的是，我们可以通过这本书，触摸到作者的思想，学习作者的做事方法和设计技巧。这就好像我们去“渔家乐”品尝新鲜美味刚钓上来的鱼，还偷师学会了钓鱼的技巧。今后想吃鲜鱼的时候就可以自己动手丰衣足食了，真是一件不亦乐乎的快事！

这本书中有些带有个案性质的经验教训也许不具备普适性，当外部环境发生变化，新的技术或产品特性的涌现，都有可能让一些经验的正确性发生“漂移”或“翻转”。但是作为作者真实的实践历程和心得体会，它的参考意义是一直存在、不会过时的。

中国的传统技术类图书，总是给人严谨有余，活泼不足的感觉。传统图书的作者在写书时往往更注重理论知识的系统性和正确性，而忽视了实践性、创新性和可读性。这也许与填鸭式的教育体制本身相关联。老师教什么，学生就学什么；作者写什么，读者就读什么。这样的一个缺乏互动的局面，已经滞后于技术

发展的时代需求。

相比之下，网络上的众多草根派高手，他们思路开阔、实战经验丰富，并且乐于互助。在网络上，作者和读者的界限变得不再那么清晰，他们以博客和论坛为交流平台，每个人都可以发帖子、写技术博客，与别人分享自己的经验，同时又从他人那里吸取技术的养分。

网络交流已经渐渐崛起成为一股新的力量。当这股新生力量遭遇传统图书时，必将如摧枯拉朽般打破僵化的图书出版局面。变革悄悄来临，谁能驾驭变革，谁就能成为胜者。

匠人很高兴地看到，北京航空航天大学出版社的胡晓柏先生及时把握了时代的脉搏。他瞄准这一新的市场趋势，及时地推出《博客藏经阁丛书》，邀请网络上的大虾一起参与技术图书的创作。这套丛书已经取得了不错的市场业绩。

《感悟设计：电子设计的经验与哲理》就是在这样的背景下，从作者的私人硬盘上被挖掘出来并加以整理后呈现给读者们的。匠人相信在未来，会有越来越多的介绍实战经验的图书与读者见面，并且匠人相信这种原汁原味的设计笔记式的书也一定是读者们所喜欢见到的。

关于这本书的内容，作者已经在自序中做了一些介绍，匠人就不再重复了。匠人最后想说的是：打开窗户，才能把新鲜空气放进来。开窗户的那个人应当被众人感谢。

程序匠人
2008－11－28 于上海

前　言

从事电子类的工作十几年了，其中纯粹设计工作也超过了10年。每天都和软件和硬件打交道。总结这些年来，学到了一点知识，得到了一点经验，也感悟了一点东西。借着本书希望和大家分享这些东西。

本书主要从设计者的思维角度来展开叙述。但和别的讲设计的书籍显著的不同在于：本书不讲某个专题，比如单片机、C语言、DSP、数据结构、操作系统等，而是注重描述在设计的整个过程中的思维方式和解决问题的方法。

很多书都是以案例来教读者如何掌握设计中一个门类的技术。比如通过一段一段的C语言代码，语法、数据结构和流程控制语句来教编程；教你如何绘制一个铜箔、一个过孔和建立元件库来绘制电路板。这些书一般不会写作者设计后的感悟。而本书和别的书不同，很多文章都是我多年来感悟出来的一些道理，这些道理和我们生活中的一些哲理一样，所以本书的一些文章标题很多是一些哲学的道理，比如"上工治未病"、"模拟，无处不在"、"物以类聚，信号以群分"和"设计的中庸之道"等。我试图通过形形色色的案例来说明这些道理。在这些文章中，都是以哲学道理为框架，中间穿插了大量的设计案例和生活案例，用以说明这些哲学论点。

为了便于理解，本书在讲述一些问题的时候，尽量避免罗列出一大堆的数学公式。因为我相信，理解繁杂的数学公式会很费脑子，阅读起来比较慢也比较困难。数学公式描述问题虽然严谨，却难免会少了几分作者的理解和感悟。所以我在表述这一类问题的时候，尽量用我的经验和感悟来说明，尽最大努力压缩数学公式的使用。同时我还在书中大量使用了插图，因为插图比文字更容易理解。

书中的很多文章，是我多年设计经验的总结。我把它们按照我自己的感悟写出来，然后按照类别分类在不同的文章里面。比如，关于电源的经验都在文章"搞定电源"里面，如何解决故障和消灭噪声的经验都在文章"搞定故障"和"搞定

噪声”里面。还有一些很容易套用的经验，比如文章“电子元件故障发生概率排行榜”和“让你的软件飞起来”用起来相信各位读者会很容易得心应手。

本书作为一本讲述电子设计的书籍，却在其中穿插了很多非电子的案例和知识点。因为在我看来，无论什么学科的知识，都是用来解决问题的。既然是用来解决问题的，那就会有很多共性，因为即使是不同学科的问题，它们本身也经常可以归纳出共同点。所以把这些东西贯穿起来，更容易感悟出事物的内在规律。

都说做设计难，很多东西只可意会，不可言传。许多设计书籍，比如教单片机的、教 C 语言的、教数据结构的和教画电路板的书籍，它们都是针对一个单一的事情来展开讨论。翻开书本，花一点功夫去消化，将书本上的内容理解，一点都不困难。甚至可以临时抱佛脚强化记忆后去应付中国式的考试。可是把这些知识拿去做一个真实的设计，却会碰到很多稀奇古怪的问题。而这些问题通常不会在那些书上有任何的描述。这个就好像是看了《三十六计》和《孙子兵法》就领兵打仗一样，虽然作战的套路写在纸上就那么多，但是什么时候该用什么套路却是一种大智慧。作战的时候，有时适合防守、而有时却要一鼓作气进攻。如果用错了策略，纵有雄兵百万也白搭。

和作战一样，做设计的时候也同样需要讲究策略，什么时候用什么策略，这种东西一般的书上不会说。因为这是一种基于实践而归纳的，游离于书本之外的知识，也就是我们常说的“只可意会，不可言传”的东西。而这些知识大多专业的成分少一些，而哲学和经验的成分多一些。本书就是致力于写出这些“只可意会，不可言传”的东西。当然，因为这些东西本身是“不可言传”的，所以文字表达起来也比较困难，因此这十分考验我的文字能力。我努力写出了一些，还有一些，我一时半会还不知道怎么表达。所以，即使是已经写出来的东西难免还有不妥和错误的地方，请各位看官多加指点和海涵。

书中的一些经验和感悟，都是站在我自己的视角，按照我的思维方式和逻辑方式去理解的。所以这些内容仅代表我个人的看法。而这个世界的思想是多样化的，看待相同问题的角度也是多样的。所以本书的内容难免存在不对和值得商榷的地方，如有不对，烦请指点。我的 E-mail：herald_book@163.com。我很愿意和各位读者在 E-mail 中探讨这些问题。

十分感谢 EDN 网站的支持，本书在 EDN 网站上开通了书友会，欢迎各位读者加入该书友会。书友会的地址是：http://group.ednchina.com:80/



1638/。

本书有幸由程序匠人作序，再次表示感谢。

最后，谨以此书献给我的母亲。是我的母亲将多年行医的经验和感悟不断传授于我。虽然我不从事医学工作，但我同样受益匪浅。因为这不仅给我带来了健康的身体，也让我在这中间感悟到了电子和医学之间的共通之处，也就是说：考虑问题的角度、解决问题的办法。同时我也以这本书献给我的兄长，因为他总是在我迷茫的时候点亮我智慧的明灯，让我能够用浅显的哲理来理解这个世界。

2008-11-29

目 录

1

模拟，无处不在

经常翻阅各种电子行业的平面媒体，总是看见 Analog Device 公司的广告，在广告画面的下方，都会有这样一个广告词“Analog is everywhere”(见图 1.1)。翻译成中文，就是本文的标题“模拟，无处不在”。

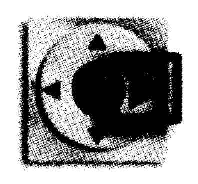


图 1.1

我想，Analog Device 公司是想通过这个广告词表达以下几层意思：

1.1 任何电路都会有可能用到模拟器件

我虽然和 Analog Device 公司没有任何利害关系。但是凭良心说，在电子行业混了十几年，自己也感觉 Analog Device 公司的模拟产品的质量还是拿得出手的。它通过这个广告词来强化“Analog Device”商标在电子行业的品牌效应。

任何电路都会有可能用到模拟器件，这句话换一种说法，就是：“这个世界没有 100%的全数字产品”。任何电子产品都会有模拟电路的部分，都有可能用到工作在线性状态的模拟器件，比如表 1.1 所列的电子产品。

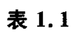


表 1.1

电子产品	说　明
MP3、MP4	公认的数字产品 但是至少它的 audio 部分是模拟的。MP4 的 TFT 屏幕对外接口虽然也是数字的，但是显示出华丽的色彩，那些色彩就是由数字信号转换而来的模拟信号
收音机、CRT 电视	传统的模拟电子产品。相信不会有人说它是数字的
平板电视	信号部分一般都是数字信号处理的。但是其前端高频头是模拟的、伴音输出是模拟的、TFT 屏幕最后还原出来的彩色信号也是模拟的
光驱、硬盘	无论是 SATA 接口还是 IDE 接口，在其前端读取数据的部分，都是模拟的。 光驱靠激光头还原出微小的光盘反射信号，放大后经过处理得到光盘上的数据；硬盘靠磁阻磁头还原出微小的磁信号，放大后经过处理得到磁盘上的数据；SATA 接口是一种电流型的串行接口，在数据发送端和接收端也都需要工作在线性状态的模拟器件，否则就不能传送高速的数字信号了。 硬盘上的信号也是模拟储存的，硬盘上面的磁阻磁头在放大信号后，经过一个判别电路，电平大于某个数值的判断为 1、小于某个数值的判断为 0。光盘上的 0－1－0－1－0－1－0－1 信号，在光盘上虽然是用一个个细小的坑洞来表示，看似“很数字”，可是由于光盘反射层的反射率有大有小、激光头的功率有大有小、盘面距离激光头的距离还会有微小的瞬时抖动，实际上激光头读出来的信号仍然是可大可小的模拟信号。需要经过信号自动增益控制电路，稳定信号的幅度，然后解调才得到数字信号
手机	中间的信号都是数字的，前端的射频信号处理和后端的声音还原都是模拟的。从我看过的 NOKIA 的 N7x 系列手机电路图和 MTK 山寨机的手机电路图来看，它们后端的音频功率放大器为了省电，全部使用 PWM 调制，做成很节能的 D 类放大器，将音频信号重放出来。这也是一种形式的 D/A 转换
以太网交换机	以太网芯片都有一个模拟前端，工作于线性状态。还有一个以太网变压器，上面传递的其实也是模拟的电流环信号。数字信号经过模拟编码后，通过电路板上的以太网变压器传送。以太网前端还要考虑变压器和线缆之间的阻抗匹配。这些都是模拟电路里面经常要考虑的东西
U 盘	USB 口看似一个数字接口，其实它和以太网一样，前端也是模拟的电流环信号。USB 口有 4 根线，分别是＋5V、GND、DATA ＋和 DATA －。其中 DATA ＋、DATA －构成了一个模拟的电流环信号。
台式机 CPU	CPU 处理的都是数字信号，但是随着 CPU 工作频率越来越高，其设计也越来越讲究。最早的 CPU 其地址总线和数据总线都是一个个的 TTL 信号。但是随着 CPU 工作频率的升高，这种方式变得很不稳定。而改进的措施，都是基于模拟技术而改进的。正是由于这些模拟技术的存在，才让这些信号传输又稳定、又高速。 随着 CPU 工作电压的下降，电源的噪声和半导体材料的噪声对 CPU 的干扰越来越明显，为此 CPU 内部的信号传输都作了很多变化。而无论是高频的无线电发射电路，还是高频的 CPU 电路，设计的过程都很类似

1.2 凡是电路，其实都是模拟电路

我做了这些年的设计，慢慢有些感悟。其中一个感悟就是："这个世界其实没有数字电路，所有的电路都是模拟的。数字电路只是模拟电路的特殊形式，正如同正方形是矩形的特殊形式、圆形是椭圆的特殊形式一样。"或者更简单地说："数字电路是模拟电路的一个子集。"

这两句话可能比较难以理解，首先来理解"正如同正方形是矩形的特殊形式、圆形是椭圆的特殊形式一样"这句话。回头翻阅一下几何书，书上说，一个矩形，如果它的长度和宽度一样，那它就是正方形，所以正方形是矩形的特殊形式；一个椭圆，其长轴和短轴的长度相等，那它就是一个圆形，所以圆形是椭圆的特殊形式。

"数字电路是模拟电路的特殊形式"，就比较难理解了。

为了说明这个问题，我找了一个 TTL 集成电路 74HC08 的规格书，相信没有谁会否认这个 IC 是数字电路吧？图 1.2 是这个 IC 的直流参数表。其中一项 V_{IH}(高电平输入电压值)有最大、最小和典型值三个数据。在不同的电压下其数据还不一样。比如 IC 供电电压 $V_{CC}=6$ V 的时候，最小值为 4.2 V、典型值为 3.2 V。就是说，要让这个 IC 认为输入的信号是高电平，一般要求输入电压要达到 3.2 V，在最坏的情况下，也要达到 4.2 V。达到 4.2 V 它绝对就可以 100％认可你的输入电平为高电平。同样的道理，IC 供电电压 $V_{CC}=6$ V 的时候，V_{IL}(低电平输入电压)至少要小于 2.8 V。

那问题就来了，在 2.8 V 和 3.2 V 之间的输入电压，算 0，还是算 1？答案是：这个 IC 会随机的认为这是 0，或者是 1。

把以上问题用图示表示，如图 1.3 所示，就能更好地说明问题。

- 如果要让 IC 的输出和输入百发百中确定地相关，就要把输入电压控制在 A 和 E 的范围内。
- B 和 D 的范围也许可以稳定工作，但是未必能够 100％保证可靠，也许不可靠的概率可能有 10％，也可能只有 1％，比较难发现。但是电路就埋下了一个爆炸时间为未知数的炸弹。
- 如果不幸把输入电压放在了 C 的范围内，此时这个 IC 就判断不清楚输入的电压是 0 还是 1，所以在这种情况下，出现问题的概率非常大，也许



DC CHARACTERISTICS
Family 74HC08
At recommended operating conditions;voltages are referenced to GND(ground=0 V).

SYMBOL	PAPAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		OTHER	V_{CC}(V)				
T_{amb}=25 ℃							
V_{IH}	HIGH-level input voltage		2.0	1.5	1.2	–	V
			4.5	3.15	2.4	–	V
			6.0	4.2	3.2	–	V
V_{IL}	LOW-level input voltage		2.0	–	0.8	0.5	V
			4.5	–	2.1	1.35	V
			6.0	–	2.8	1.8	V
V_{OH}	HIGH-level output voltage	V_I=V_{IH} or V_{IL}					
		I_O=–20 μA	2.0	1.9	2.0	–	V
		I_O=–20 μA	4.5	4.4	4.5	–	V
		I_O=–4.0 mA	4.5	3.98	4.32	–	V
		I_O=–20 μA	6.0	5.9	6.0	–	V
		I_O=–5.2 mA	6.0	5.48	5.81	–	V
V_{OL}	LOW-level output voltage	V_I=V_{IH} or V_{IL}					
		I_O= 20 μA	2.0	–	0	0.1	V
		I_O= 20 μA	4.5	–	0	0.1	V
		I_O= 4.0 mA	4.5	–	0.15	0.26	V
		I_O= 20 μA	6.0	–	0	0.1	V
		I_O= 5.2 mA	6.0	–	0.16	0.26	V

图 1.2

电路立即就会死得很难看。

把以上问题用如图 1.4 所示的坐标图表示,更容易理解了:

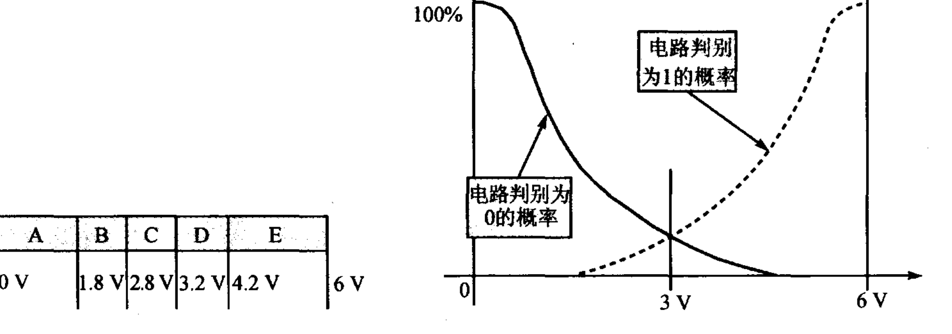


图 1.3

图 1.4

看明白了吧！即使是数字电路，其输入输出其实也和电压的大小强弱密切相关，多一点或少一点都可能造成不同的结果。

传递数字信号，比如在远距离和高吞吐量的数字传输方面，也经常用模拟技术来传递平常所说的“数字信号”。这个方面最好的例子，就是电流环的 RS485 信号。大家都知道 RS232 接口的信号不能传递很远，而 RS485 接口的信号就可以轻轻松松地把信号传递到 3～4 km 外。RS232 信号之所以不能传远，就是因为它将 0－1－0－1 的数字信号转换成了＋10 V、－10 V 的电压，用电压通过电缆进行传送。这些电压在信号的发送端都可以做得很好，但是通过电缆之后，由于电缆的分布电容、分布电感、空间的干扰、信号的振铃和反射等因素，到电缆的对端接收后，信号都会有所变形。而且电缆越长变形越严重。

为了解决这个问题，使用电流环差分信号来传送数字信号。

图 1.5 是常用的 MAX485 芯片的信号传输等效图，左边的驱动器 D 用于发送，右边的接收器 R 用于接收。2 个 100 pF 的电容用于模拟信号电缆的分布电容。54 Ω 的电阻是信号匹配电阻。

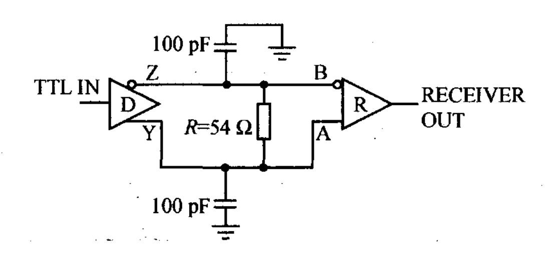


图 1.5

发送器的输入 TTL IN 就是图 1.6 里面的 DI，RECEIVER OUT 就是图 1.6里面的 RO。图 1.6 里面的 $V_y - V_z$ 是发送器的 Z 输出和 Y 输出之间的电压差。由图可见，它有点像一个锯齿波，而且摆幅也在±1 V 之间，已经是公认的模拟信号了。由图 1.7 可见，在接收端的 A、B 端点之间，只需要电压差的绝对值大于 0.2 V 接收器就可以接收，这也说明，IC 内部其实是有模拟电路的，起码，需要将 0.2 V 的信号放大到 2～3 V 的水平。

从这里，可以看到在传输路径中，经过了 2 次信号变换，分别是“数字信号”→差分模拟信号→“数字信号”，而在线缆上传送的只是差分模拟信号。

使用以上方法，可以大幅度减少噪声对电缆的干扰，也可以最大限度的弱化电缆直流电阻、分布电容和分布电感的影响。这就是为什么 RS485 的传输距离

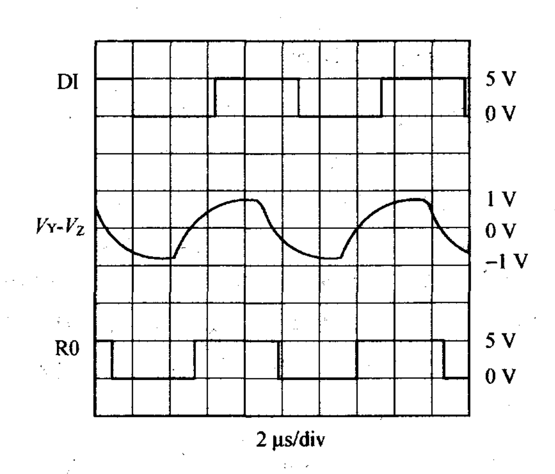


图 1.6

Receiving

INPUTS			OUTPUT
$\overline{RE}$	DE	A-B	RO
0	0	≥+0.2 V	1
0	0	≤-0.2 V	0
0	0	Inputs open	1
1	0	X	High-Z*

X=Don't care
High-Z=High impedance
* Shutdown mode for MAX481/MAX483/MAX487

图 1.7

远远高过 RS232 接口的缘故。

类似的原理,电流环还应用在 LVDS 传输接口、1394 总线、以太网接口、SATA 硬盘传输线和 USB 传输线上。LCD 电脑监视器屏幕的信号电缆也是用这个原理传输的,所以一个小小的电缆可以轻松传送监视器那样的高带宽信号。

本书中还有一篇文章《搞定噪声》里面的章节"注意你的高速数字信号",如果您能理解这个文章所陈述的思想,就会发现,传送高频率的数字信号其实很麻烦,不仅有传输距离的限制,还会有振铃、反射等一大堆的问题。这些问题都是

模拟电路中才有的问题。而要传输好这些高频率的数字信号，有且仅有使用模拟的这些解决之道才能搞定。

与文章《搞定噪声》里面的章节“注意你的高速数字信号”的原理类似，读者可以拔一条计算机的内存条出来看看，上面除了内存颗粒，还有很多贴片的电阻，而且大都靠近连接器的位置，这些基本都是降低振铃和反射等问题用的（见图 1.8）。



图 1.8

所以，设计一个计算机主板，其实是一个很有技术含量的工作。市场方面有成本压着，不能随便使用高档材料，技术上，CPU 系统的工作频率不断升级，电路的布线难度越来越高，还要考虑散热、电磁兼容等一大堆问题。所以虽然很多人认为设计电路板是个简单劳动，可是设计主板的工程师绝对不会是菜鸟。

1.3 推而广之

刚参加工作的时候，经常骑自行车，由于有自己的习惯，喜欢自己维修自行车。记得当时把前轮拆下来修理后，以为自行车没什么大不了的，就随随便便地把前轮装上去。虽然也能骑，却骑得很不舒服。后来拿到维修店请师傅看，才知道，车轮在安装的时候，车轮和主支架有夹角，所以车轮转动后就不平衡了，轮胎会左右晃动。维修店的师傅又把它拆下来，仔仔细细地慢慢调正，前轮的主轴螺丝被一点、一点地调紧，一边调整还要一边不时地转动轮子，看看轮子转动后的动态平衡状况。之后自行车才有了比较好的使用感觉。

类似的,还有自行车的链条。调紧了,骑的时候很费力(传动效率低),链条本身发生金属疲劳的可能性也会增大许多,金属件发生金属疲劳通常就意味着会有裂缝或即将断裂之类的毛病。调松了,发生脱链的概率就大幅度上升。把它画成坐标图好理解(见图 1.9)。可以看出维修店师傅的卖点就在于他能准确把握这个最佳工作点。把车子调整到这个工作点就需要费一些时间,边调边观察车子的各种状态,最终将自行车调节在这个最佳的工作点上。从这个角度看链条和调节前轮主轴的例子,和调试模拟电路工作点不是如出一辙吗?

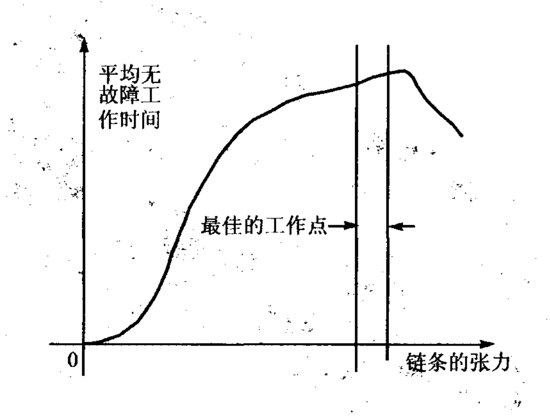


图 1.9

同样的,在医学上,健康和生死问题也不是非黑即白的,在病与非病之间、生与死之间也不是界限分明的。

首先说说健康与生病之间的关系。很多人认为,一个人没有生病就是健康,不健康就是生病。其实在健康和生病之间还有中间状态,就是所谓的亚健康状态。

图 1.10 和表 1.2 给出了人健康状态的描述。

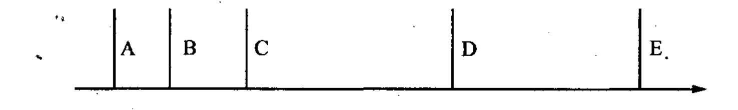


图 1.10

表 1.2

状 态	描 述
A点	绝对健康。有非常好的免疫力，即使天塌下来也不会生病。可惜这种人太少了
B点	小病，如口臭、疲劳、困倦、脱发、皮肤起红疹等
C点	通常所说的"生病"是指从这里开始的，比如口腔溃疡、感冒、腹泻。得这些病之前通常都有B点的症状之一
D点	大病，需要输液或者住院，N天到N个月才能恢复的病
E点	绝对的不健康，随时都可能转入死亡的大病，如糖尿病晚期、肝癌晚期等。在这个状态的，都已经元气大伤，并且回到健康状态的概率很低，治疗的成本也很高

有些人会突然出现E点的症状，但如果患者仔细观察自己身体的话，其实至少B点的症状在E点症状出现之前至少会有一些。这个例子就说明病与非病、健康与非健康之间也和模拟电路一样，是有渐进过程的。

生与死的界限同样也是有过渡过程的。曾经就见过一位古稀老人，因为消化道急性炎症，刚开始还能喝点稀饭之类的。因为炎症没有控制住，后来只能喝水，身体也日渐消瘦，靠输液支持着。再往后，因为营养的问题，出现并发症且意识模糊，在最后的40小时左右，说了最后一句话。这个时候医生已经告诉家属，该患者已经彻底没有希望了。最后的几个小时只能嘴唇微动，但可以听见周边人的说话，还会用眼神对别人的话作出反应。在她彻底失去反应之前，手一直是热的。在最后阶段，你不能说她已经死了，但是她明显也不像是个大活人。从生病到驾鹤归西，整个过程历时两个月。这个例子也说明生与死同样也不是非0即1的。

即使所谓的"死亡"真的发生了，比如医生宣布某人脉搏和呼吸都停止了，并且脑死亡也已经发生了。但是很多人不知道的是：其实这个时候身体的很多器官还在运转。《法医学》上的描述是：如果脑死亡发生20分钟之内，很多人体器官还有可供移植的价值，2小时之内肠子还会蠕动，4小时之内人体的某些肌肉对于一些化学刺激还有反应。这个例子同样也说明，即使是死亡真的不可逆转地发生了，其过程同样也是渐进的，是一个"模拟"的过程。

经常可以看见一些菜鸟工程师，在画单片机电路板的时候，简单地认为这是一个纯数字的系统，只要把线路全部连接起来就好了。事实上把线路全部连接起来，确实会有一些电路真的可以安全工作。但是一些稍微复杂的电路，只要碰到以下几种情况之一，就可能会出问题：



- 工作频率很高的数字电路,比如高速 CPU 系统。
- 大功率的系统,比如开关电源。
- 电源上叠加了比较大的纹波。
- 信号线比较长。
- 工作电压比较低,比如 1.8 V 的系统。
- 微弱信号放大电路,比如硬盘的磁头信号放大器和光盘的激光头信号放大器。

所以,我的设计习惯是:无论什么电路,都把它看作模拟电路去设计,认真对待每一个铜箔的接线、认真计算每一个接口的阻抗和电平;出现异常的时候,用示波器仔细分析波形是否变形。只有这样,数字电路才能有足够的稳定性。

提示:

- 生活中也有一些非 0 即 1 的东西,比如彩票 xxxxxx 号码可以中奖 500 万,差一号就一分钱拿不到。这些东西通常是人类为了便于管理某项事物而人为制定的游戏规则,并非事物的本源。
- 理解了所谓"数字电路"的模拟特性和生活中的"模拟",你就不会按照非 0 即 1 的思维去理解数字电路。
- 把任何电路都按照模拟电路来设计,重视并且控制"数字电路"里面可能发生的"模拟"问题,是保证硬件电路稳定性的重要一环。

2

上工治未病

做研发和设计，就是一个发现问题、解决问题的过程。如果问大家："发现问题、解决问题的最高境界是什么?"您会如何回答呢?

如果我来回答这个问题，我的答案是：要学会如何避免问题发生，知道如何避免麻烦出现，才是解决问题的最高境界。因为解决问题都是有成本的。比如要付出时间、精力和金钱等。这个道理类似于兵家的一个问题：用兵的最高境界是什么？不是如何研发新武器，也不是如何进攻、如何防守，而是一句很经典的话"不战而屈人之兵，善之善者也。"真的佩服中国古代先人的智慧，因为先人也说出了很简单的理由："用兵一日，日耗千金"。所以不打仗能战胜敌人当然是行军打仗的最高境界了。

以我的观点来看，无论是电子设计、人体健康等方面，在出现大问题之前一般都会有小的征兆出现。如果能发现这些小的征兆，并加以控制和扑灭，就可以非常有效地降低大问题出现的概率。这个道理可以用一篇我们都很熟悉的古文来说明：

扁鹊见蔡桓公

韩非子

扁鹊见蔡桓公，立有间，扁鹊曰："君有疾在腠理，不治将恐深。"桓侯曰："寡人无疾。"扁鹊出，桓侯曰："医之好治不病以为功!"

居十日，扁鹊复见，曰："君之病在肌肤，不治将益深。"桓侯不应。扁鹊出，桓侯又不悦。

居十日，扁鹊复见，曰："君之病在肠胃，不治将益深。"桓侯又不应。扁鹊出，桓侯又不悦。

居十日，扁鹊望桓侯而还走。桓侯故使人问之，扁鹊曰："疾在腠理，汤熨之

所及也；在肌肤，针石之所及也；在肠胃，火齐之所及也；在骨髓，司命之所属，无奈何也。今在骨髓，臣是以无请也。”

居五日，桓侯体痛，使人索扁鹊，已逃秦矣。桓侯遂死。

以上这篇短文的作者韩非子还留下了一个结论：

故良医之治病也，攻之于腠理，此皆争之于小者也。夫事之祸福亦有腠理之地，故曰："圣人早从事焉。"

如果用数学的方法，把韩非子的这篇短文归纳一下，可以用一个1维坐标线来大致表示人体的状态，将人的健康从0到100划分成不同等级，其中100%健康代表百毒不侵、什么病都没有的状态；0%健康代表已经完全没有了生命体征，即死亡的个体，如图2.1所示。

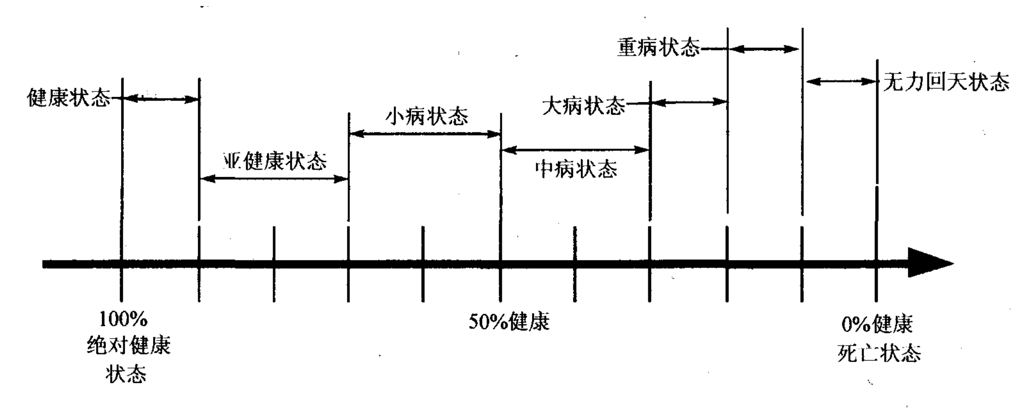


图 2.1

- 扁鹊曰："君有疾在腠理，不治将恐深"，代表病人在亚健康状态。这个时候大概桓侯身上出现了一些典型的亚健康状态的表现，被扁鹊观察到了。而桓侯大概是由于对医学知识的缺乏，或者是他认为这些根本就不是问题，很多人身上都有。所以桓侯曰："寡人无疾"。还讽刺扁鹊说："医之好治不病以为功！"。就这样，他错过了解决问题最好的一个时机。这个时候只要将人体做一个小小的休整（比如多睡觉、多喝水、喝一点汤药等），成本极低，时间也很快。
- 扁鹊复见，曰："君之病在肠胃，不治将益深"，代表病人在中病状态。这个时候，在扁鹊看来，已经有一些比较大的症状了。可能这个时候桓侯

身上的症状还只是专业人士用专业眼光才能看出来，还没有到桓侯这个外行能感觉到"病了"的程度。此时，治疗的成本比"疾在腠理"要高一些。不过治疗好的概率还是相当高的。由于桓侯不重视细节，他丧失了这一次，也是最后一次可以痊愈的机会。

● 扁鹊望桓侯而还走，代表病人在重病状态。在扁鹊看来，这个时候桓侯的病已经到了治疗成本极高，治愈概率极低的时候，所以他"望桓侯而还走"。

通过这个故事说明：很多大问题在爆发之前，一定会有小问题，关键在于你能不能发现和捕捉到它们。如果你足够的细致，就能捕捉到很多这样的细节。在大问题的前兆阶段，体现出来的问题就是一些小小的异常。这个时候如果能及时处理，成本极低。而一旦问题没有被发现，或者你侥幸地认为"这个问题可能不是问题"，那一旦大问题的风险兑现后，你就要付出高昂的成本来解决问题。有时候甚至问题都大到神仙也无力回天的程度，那个时候只能眼睁睁地看着问题朝向不利的局面发展。比如前面所说的"居五日，桓侯体痛，使人索扁鹊，已逃秦矣。桓侯遂死。"

人的生命历程，就是一个在图 2.1 坐标线上左右震荡的过程，而震荡的大趋势是朝向 0%健康度的方向。而做产品设计，则刚好相反。刚刚做完的产品总是不可靠的，需要我们不断的发挥智慧和勤奋，它的可靠性才会越来越好。虽然有这样重大的差别，但"上工治未病"这个至理名言在设计中也同样成立。

2.1 大问题，是因为根基没有做好

这些年一直有中医存废之争。我是坚决支持中医的。因为我好几次生病，花了很多人民币用西医的方法治疗，没有多大效果；但是用小钱找中医却把问题解决了。以前不太明白这其中的道理(到现在也不十分明白)，但我觉得，它能把问题解决，就说明它是合理的。而且我不止一次通过中医治疗好了疾病，所以就不存在中医瞎蒙之类的偶然性。

去年有幸在福州的香格里拉酒店听了一位北京来的医学大家的演讲，他的几句经典归纳，说出了中医好在什么地方。他说：

● "我们的中医和中药是从宏观和哲学的角度去探讨人与自然、人体内在的和谐，强调的是整体诊疗观；而西医则是实践经验学，通过对局部细节的深入分析，理解和认识人体规律"。

● "西医的治病，重在解决眼前的问题，所以它对急性的病症很有疗效，却

对系统性的疾病(比如亚健康、林林总总的慢性病等)效果不大。”

- “中医正好相反,它通过宏观的视角,提升人体内在的潜能,让人体各个构件更好的磨合来治疗疾病,通过修正人体某些宏观上的不足来降低疾病的发生概率。”
- “中医最精妙的地方,在于它如何防止人生病(即降低疾病的发生概率)。治病并不是中医最精妙的地方。”

假设有 A 和 B 两个人。B 有甲型肝炎且自己不知,而 A 也在不知情的情况下和 B 握手,结果也感染了甲型肝炎。那这个时候,谁是 A 生病的第一责任人呢?

通常情况下,绝大部分的人会说都是 B 害的。可是我的理解是:因为 A 在一次生活中很常见的握手动作中患病,说明 A 没有足够的稳定度(即足够的健康),所以才容易受外界的变化而产生故障(患病)。所以 A 才是第一责任人,B 只是在 A 不够稳定的时候,触发了他而已。而这种触发本身并没有不合理的地方。因为 A 自己的不稳定,所以即使他不碰见 B,而碰见患有 SARS 的 C,或者碰见患禽流感的 D,都有可能被感染。而如果 A 平时有充足的体育锻炼和充足的睡眠,那他基本就有足够的健康程度,再碰见这些病人的时候,他患病的概率就会小得多。

这个世界虽然有很多的不足和不合理,但是有时候改进自己,去适应这种不足,比去试图改造这些不足更合理。

前几年有熟人拉我去诊断一个他们设计的绣花机的工业控制系统,他们说,这个系统从诞生以来就毛病不断,可靠性非常差。我去了之后,看过他们的机器实物,调阅他们的图纸和软件,发现了很多问题:

硬件方面:

1. CPU 控制部分和 220 V 的开关电源设计在一个电路板上。这样做的缺点是难以生产。220 V 的 AC/DC 电源模块在电子市场一抓一大把,像这样没有多少产量、售价高的系统,自己去做电源,给人的感觉就是造飞机的人还要自己去设计螺丝钉、铆钉一样。不仅增加了采购和制造的难度,也增加了电源带来故障的隐患。因为自己做的小批量电源,其可靠性很难超越市面销售的、大量生产的品牌电源的可靠性。
2. 用示波器看,CPU 电源的纹波很大,达到 150 mVpp,这个也能降低硬件的可靠性。
3. CPU 电路板和电动机的控制电路板之间采用 SPI 端口连接,两个电路

板之间用一个 1.5 m 的电缆连接。CPU 和外设通信用 SPI 端口不是不可以，但是电缆这么长的话，很容易出现本书"搞定噪声"一章中所提及的"注意你的高速数字信号"讲到的问题。（他们自己也讲，有时候通信不可靠）虽然用我说的方法可以解决一些问题，但我的方法并不能对付一个 1.5 m 长、而且跑高速数字信号的 SPI 电缆。如果要解决这个问题，必须重新设计硬件接口。

4. 电路板上很多输入输出口上，都没有 ESD 保护，这样也会大大降低硬件的可靠性，尤其在插拔连接器的时候。
5. 有一些电机控制的机械部件，总是不能精确定位，虽然加入了一些传感器后，CPU 多次调整后可以达到精确定位，但是付出了时间的代价，也就是降低了机器的生产效率。而我认为像这样的地方，就不应该使用普通电动机，而应该使用步进电机来设计，这样控制起来就简单多了。
6. ……

软件方面：

1. 主控制电路板的单片机软件，一共 4 000 多行，没有按照模块分割，全部写在一个文件里面。看这个软件，第一感觉就是"晕"。
2. 公共变量比较多，一般像 4 000 多行的软件，公共变量在 10 个以下的，还基本算正常，它居然达到了 160 多个。
3. 软件的注释极少。
4. 软件的变量名和函数名很多是采用没有意义的字母或者拼音，晦涩难懂。
5. ……

像以上硬件方面的问题 1、3 和 5 都属于在项目策划阶段的错误，剩下的问题属于项目执行阶段的错误。项目执行阶段的错误修正起来相对还比较容易，可是项目策划阶段的错误基本上就属于"路线错误"了，修正起来难度极大。

以上这些问题，说起来每个问题都是小问题，对可靠性造成的影响都不大，但是一大堆这样的问题，就能造成基础性的大问题。这些问题的出现，在于他们平时对研发人员没有进行有效的疏导和管理。所以，对这个机器的修改，基本上要对这几个电路板推倒后重新设计，工作量不小，无论是时间进度还是工程造价的要求都无法满足对方，我也只能学古人"望桓侯而还走"。

2.2 大问题一旦爆发，就很难处理

我在一个医学网站上看到一个案例，这个案例是讲某人的母亲癌症爆发后的处理过程，有点借鉴意义。原文如下：

我家的老人上个周六在痛苦的挣扎中，走完了她生命的最后旅程。

在本组中，我曾咨询过尊敬的jerry与edward大夫，感谢您二位素不相识的朋友，曾给我的指导与建议。本来我想取消本组的订阅，因为痛苦的历程与记忆太过于深刻的留在我的脑海，但为了给2位大夫提供反馈以及与其他朋友提供相应的经验，还是特此写了本贴。

在去年的7月18日，那是著名的唐山7.18大雨之夜，我母亲突然发高烧并憋闷，我深夜才回的家。与家人商谈后，决定有限度的告诉我母亲她的真实病情，令我欣慰的是她并没有如我想象的那样惊恐失措，反而是平静的询问我下一步的治疗计划。这才让我下定决心去肿瘤医院，我身边的朋友早就劝告过我，治疗肿瘤最好去专业医院。

住进肿瘤医院后，CT检查胸腔有积液，在排出了近2 000cc积液后，呼吸憋闷很快改善。检查身体后，发现白蛋白水平极低，做注射人血白蛋白补充。随后腰椎与肋骨开始了剧烈的疼痛，随即开始放疗，在放疗期间有段时间并无医生所判断那样疼痛减轻，差点中途放弃。但最终止疼效果明显。过了这艰难的一关。在此同时，CT检查发现我母亲颈骨有破坏，导致左手活动障碍，同时显示右颈淋巴开始肿大。大夫说下次准备做这2处的放疗。

随后回家休息了近20天后，吃的中药“参芪胶囊”与西黄丸，并没有什么疗效。重新住院后，我母亲突然感到腹部剧烈的疼痛，但全身的增强CT显示出了原来的骨转移与肺部转移外，并没有其他脏器的转移。感谢我所托付的主治医师，在他们多次商讨后，认为是因为T10.T9脊柱的原因导致，随后也对这2处放疗。

在骨转移的疼痛有效制止后，不幸的是肺部的肿瘤突然剧增，在一个月的期间内，由2 cm长到了10 cm。我咨询主治医生易瑞沙是否有效，她告诉我因为原发不明，此肺部的肿瘤应该对易瑞沙不敏感，但可以一试。我就写E-Mail给印度的制药厂，药厂给我了回复，因为知识产权的原因，他们的药不能在中国卖，让我找他们在印度各地的代理商。但由于障碍太多，最终还是在国内渠道买的。

但吃完后也如医生预期那样，没有效果。在无奈之下，主治医生只好实施了单药健择化疗。在此期间，在我的坚持下，医院还是给做了肺部的穿刺，病理显示是“腺癌”。

在07年11月20日左右，我母亲突然病情恶化，那一夜心跳曾一度达到200多，值班医生告诉我，我母亲已经进入了衰竭状态，我含泪签下了病危通知书。在医生与护士整晚的治疗下，强心针与利尿剂发生了作用，我母亲奇迹般的跨过了这个门槛。

随后肺部的肿瘤急转直下，在11月29日的CT检查中，肿瘤基本上已经长满了整个肺部，呼吸完全依靠氧气支撑。医生告诉我，她随时都有可能离我们而去。在12月末时，我母亲甚至连去做CT都不能去了。在主治医生认为单药健择化疗效果不明显的情况下，改用了“歧星”作为化疗药物。在今年1月初时，又出现过一次病危抢救。

此时我无意间看到了中央4台“中华医药”栏目介绍的江苏某地的老中医，曾救活过濒临死亡的病人。在无路可走的情况下，我拍了我母亲面相与舌苔的照片，并找我当地的中医写了脉案，火速赶到了那个医院。医院的院长在看了资料后，告诉我这是严重的“气阴2虚”。并开具药方。

在近4个月的放疗与化疗后，我母亲体重已经剩到30 kg左右，而且面目几近变形，气色吓人。在抱着试试看的态度下1月5日起辅以中药，在喝了3天中药后，结果出现了我意想不到的情况：首先是全身的浮肿明显减退，也停止了掉头发。慢慢的饮食与睡眠状况均有很大改善。在今年1月末时，面色已经恢复到去年7月份的样子。最明显的改善是身体化验指标的全面改善，尤其是白蛋白水平几乎是正常值，在做了一个月的“歧星”化疗后，在没有注射升白药物与升血药物后，白细胞值还稳定在8 000左右。除了缺钠外，各项指标都显示正常，而且肝、肾等脏器功能并无损伤。

情况都在一天一天好转，监护已经减到普通监护水平。在一月末时，我母亲已经能脱离氧气，用轮椅推去CT室做检查。CT检查虽比2007年11月29日有所增加，但医生告诉我应该是“歧星”化疗有了作用，肿瘤最大值应该是在一月初，相对最高峰时，肯定有所减少，同时她告诉我我母亲下咽部的肿块，随着化疗也减小了，有可能下咽就是原发灶。

在本月初，医生已经乐观的判断我母亲可以回家过年了。2月4日我接她出院，我们全家最后一次一起过了团圆年。

但在正月14,我母亲有些感冒并突然感到肋骨剧烈疼痛,我随即联系住进了医院。医生发现在肋骨处已经有肿块凸出,遂安排放疗。但不幸的是我母亲此时因痰液过于粘稠,在上周六下午无法自己吐出痰液,在吸痰效果不明显的情况下,很快就陷入了衰竭症状。虽2次逃过,但这次奇迹并没有发生,我就这样眼睁睁的看着监视器终于出现了直线。

就这样,一个小小的感冒引发的呼吸道感染,就让本已逆转的状况这样急转直下。我母亲就这样解脱了,在近1年4个月与癌症的抗争中,她忍受了那些令我与我家人揪心的病痛的折磨,终于解脱了。

OK,故事讲完了。我从这个例子感悟出的道理就是:越接近危险的东西,要反过头来离开危险,所需要的成本越大。从宏观而言,当一个行动执行起来的成本越大,其成功的概率也越低。

以上案例中,这个肺癌患者的健康状态,从疾病暴发之日起一直在图2.1的30%~10%之间游走振荡。前面一直扛了一年多,也没有回到比较健康的状态。而到最后的无力回天阶段的时候,只用了几天时间。本文的作者明显是一个孝子,虽然他十分努力,但是自然规律不会因为他的孝顺而有任何的变化。根据一些医生的说法,像这样的疾病,在爆发之前不可能一点毛病也没有,或多或少都会有一些异常。换一个角度说,如果他母亲更细致一些,或许就能早几个月觉察到自己身体的异常,也许就不要花这么大的成本来解决这个问题了,或许就能多活几年。

像上面提及的“绣花机的工业控制系统”的例子,也是一个大问题出现后,需要付出高成本来处理的例子。

前几年带过一个徒弟,让他写单片机软件,通过2个月的努力,终于做好了一个电表自动校正系统。但是他却在最关键的时候离职了,把我郁闷得不行。他走的时候,系统基本能用,但还有很多小毛病和不可靠的地方,需要进一步完善。在他走之前,我基本偏向于这个机器的硬件设计和整个系统的策划执行,软件基本就全权交给他实现。他走后,因为职责所系,我开始深入软件部分,尝试着去修正那些小毛病和不可靠的地方。但是随着对源代码的深入了解,越来越发现他的代码很混乱,主要体现在以下几个方面:

- 单片机软件,一共3 500多行,没有按照模块分割。全部写在一个文件里面。和前面的例子一样,这是菜鸟写软件的基本特征。
- 有时候一个子函数长度超过150行,甚至有300行一个的子函数。
- 还是和上面的例子一样,公共变量出奇的多,超过了110多个。

- 软件基本没有注释。
- 函数和函数之间的变量传递，极少使用函数的参数。大部分都是用公共变量进行传递。
- 多次使用的常数，不用＃define来定义，而是直接写在代码里面。这样修改一个参数要改很多个地方。

以上这些问题，能够影响以下2个方面：

- 程序的模块化。
- 程序的结构化。

但是，这些足以降低程序的可读性。在我看来，这个程序的可读性非常差，我相信，即使对于作者自己而言，可读性也不会太好。依据我的看法，无论是软件源代码还是硬件的原理图，降低了可读性就会提高出差错的概率，而且一般可读性越低，运行起来后出差错的概率就越高。

为了修正那些小毛病和不可靠的地方，我就必须阅读和理解这些代码，为了做到好阅读，我做了以下工作：

- 把原先单个文件的源代码拆分成 N 个，按照源代码的功能划分成独立文件。比如LCD显示屏的，就放在lcd.c里面；键盘部分就放在keyboard.c里面；核心电表校正算法就放在core.c中；通信部分的软件就放在comm.c中……
- 这些文件切割好后，编译通过，算是改造的第一个阶段完成。
- 使用UE－32文本编辑器，搜索出每一个全局变量和调用这些变量的地方，然后通过函数间的参数传递，尽量去掉这些全局变量。通过这个努力，将全局变量从110多个降低到6个。
- 在降低全局变量的过程中，需要不断修改，不断编译，以确保修改后，没有新的语法错误。
- 如果某个子函数的长度超过50行，则尽量将其拆分成多个子函数。
- 将一些具有相关性的多个变量，打包成为数据结构。
- 尽量用＃define来替代程序里面的常数。

这样，虽然我最终按照可读性的要求基本改好了这个源代码，源代码的可读性比从前好多了。软件的可靠性也有了大幅度的提高。原来一些小毛病现在都很容易修正了。但是我也付出了20个工作日的代价。之后在实验室的调试中，顺利过关。但是拿到工业生产现场，却出现通信有误码。有时候通信命令发出后，对方没有响应，虽然概率不高，可是好歹也算个问题。这下，就只好将原先切

割出来的通信部分的软件 comm.c 拿出来再次研究。通过对这个模块的阅读，发现这个模块的通信协议过于简单，根本没有考虑接收误码，没有考虑通信出错后如何继续保证通信的可靠性等问题。为此，不得不将这个模块全部推翻，设计新的通信包数据结构和算法。将他原先简单的通信流程，改造成具有 2 个通信层的代码。分别放在 comm_link.c(链路层)和 comm_protcol.c(协议层)里面。将原先的那个 comm.c 废弃。

改造后的通信模块和原先的代码链接起来后，通信的可靠性得到了大幅度提高。但是我为此又额外付出了 15 个工作日的代价。加上原先的开销，返工一共是 35 个工作日。

通过这个工程事例，使我明白，要保证可靠性，必须时时去监控项目参与者的工作成果(包括半成品)，这样就不至于出现前面的(开发阶段 2 个月，返工 35 个工作日)的情况，也就是下面所说的"及时扑灭小问题"。

2.3 及时扑灭小问题

前几年做过一个嵌入式系统，当时由于系统运行不稳定，检查发现是由于 ARM 芯片的 3.3 V 电源的纹波增大所导致的。当时的纹波达到了 150 mVpp 大小。后来通过并电容，将纹波压缩到 75 mVpp，系统基本就稳定下来了，4～5 个样板连续运行 20 个小时也没有发现问题。但当时我还是觉得 75 mVpp 的纹波大了一些，心里还是有点放心不下。可是同事把试验结果告诉我后，并且说由于电路板面积有限已经不能再增加电容了，所以就放弃了进一步压缩纹波的想法。等这个电路板量产后，发现一些电路板还是有一点不稳定，不稳定的百分比大约 3%～4%。不稳定的电路板纹波一般都达到了 90～100 mVpp。后来解决问题的办法还是继续降低纹波。所以从这个案例看，就是当时纹波处理得不够干净，斩草未除根，导致在量产的时候出现麻烦。

在抗洪救灾的新闻中，经常会提及"管涌"这个词。"管涌"就是在发大水的时候，大坝底部的漏水。它的过程是随着大坝内部的水位上升，涌水挟带出的砂粒逐渐增多，涌水量也随着加大，涌水量增大挟带出砂粒也就更多，如将附近堤坝地基下的砂层淘空，就会导致堤坝骤然下挫，甚至酿成决堤。

所以，每当发大水的时候，政府都要组织人力，在大堤上做拉网式的巡查，尽量发现哪怕是很小的类似于泉水涌出那样的异常，如果找到就要确定是不是管涌。如果是的话，就要及时处理。这个时候的处理成本，只需要一锹土或者几个



沙袋，然后派一个人 24 小时盯守就可以了。如果管涌的水由清澈变为浑浊，说明危险等级已经扩大，这个时候的处理成本可能就要几十个沙袋、甚至几卡车的物资，此时，可能就是要调一大拨人来处理了。当然，如果这个还是没有处理好，那剩下的就是溃堤了。



3

让你的软件飞起来

做同样的事情，方法不一样，效果也不一样。于是就有了2个成语“事半功倍”和“事倍功半”。比如，汽车引擎，可以让你的速度轻松超越马车，但却无法飞行。因为汽车引擎的功率都太小。如果装上大功率的多缸活塞引擎，那飞起来没有问题，可是却不能超越音速；涡轮引擎，可以轻松超越音障，却无法飞出地球；如果有火箭发动机，就可以到达火星。但是即使是目前最强大的火箭发动机，也不能将人类送出太阳系。这就是方法决定效果的最好例子。

代码的运算速度取决于以下几个方面：

- 算法本身的复杂度，比如 MPEG 比 JPEG 复杂，JPEG 比 BMP 图片的编码复杂。
- CPU 自身的速度和设计架构。
- CPU 的总线带宽。
- 您自己代码的写法。

本文主要介绍如何优化您自己的代码，实现软件的加速。

先看看我的需求：一个图像模式识别的项目，需要识别画面上的物体，为了减小 CPU 识别物体时候的运算量，需要将 RGB 格式的彩色图像先转换成黑白图像。图像转换的公式如下：

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

图像尺寸 640 * 480 * 24bit，这里不再赘述图像的采集过程，只关心图像的处理过程。现在图像经过采集后，彩色的 RGB 图像已经按照 RGBRGB 顺序排列的格式，每个颜色一个字节，放在内存里面了。

根据需求，首先定义程序的输入和输出格式，以下是输入和输出的定义：

```
#define XSIZE 640
#define YSIZE 480
#define IMGSIZE XSIZE * YSIZE
```

```
Typedef   struct RGB
{  unsigned char R;
   unsigned char G;
   unsigned char B;
}RGB;
struct RGB in[IMGSIZE]          //需要计算的原始数据
Unsigned char out[IMGSIZE]      //计算后的结果
```

看出来了吗？我已经悄悄地完成了第一个优化。图像是一个 2 维点的集合，我却用一个 1 维数组来存储。这是因为编译器处理 1 维数组的效率要高过 2 维数组。

按照公式先写一个代码，这大概是能想得出来的最简单的写法了，实在看不出有什么毛病，好了，编译一下跑一跑吧。

```
Void calc_lum()
{int i;
    for(i = 0;i<IMGSIZE;i ++ )
    {double r,g,b,y;
    unsigned char yy;
    r = in[i].r;   g = in[i].g; b = in[i].b;
    y = 0.299 * r + 0.587 * g + 0.114 * b;
    yy = y;      out[i] = yy;
    }
}
```

这个代码分别用 VC6.0 和 GCC 编译，生成 2 个版本，分别在 PC 上和我的 ARM embedded system 上面跑。速度是多少？你可能不相信，太慢了。在 PC 上，由于存在硬件浮点处理器，CPU 频率也够高，计算速度为 20 s，而在我的 ARM embedded system，没有以上 2 个优势，浮点操作被编译器分解成了整数运算，运算速度为 120 s 左右。这仅仅只是一副图像的运算速度！

上面这个代码还没有跑，我已经知道会很慢，因为这其中有大量的浮点运算。只要能不用浮点运算，一定能快很多。那如图 3.1 所示的公式怎么能用定点的整数运算替代呢？

```
Y = 0.299 * R + 0.587 * G + 0.114 * B;
Y=D+E+F;
D=0.299*R;
E=0.587*G;
F=0.114*B;
```

图 3.1

RGB 的取值范围是 0～255，都是整数，只是这个系数比较麻烦，不过这个系数可以表示为：

$$0.299=299/1000$$

所以 D=(R * 299)/1000，于是

$$Y=(R*299)/1000+(G*587)/1000+(B*114)/1000$$

再简化为：

$$Y=(R*299+G*587+B*114)/1000$$

这一下，能快多少呢？ARM Embedded system 上的速度为 45 s，PC 上的速度为 2 s。

将 0.299 * R 进一步化简，可得 Y=(R * 299+G * 587+B * 114)/1000，这个式子好像还有点复杂，可以再砍掉一个除法运算。图 3.1 中的算式 D 可以这样写：

$$0.299=299/1\,000=1\,224/4\,096$$

所以　D=(R * 1224)/4 096，于是

$$Y=(R*1224)/1\,000+(G*2\,404)/4\,096+(B*467)/4\,096$$

再简化为：

$$Y=(R*1224+G*2404+B*467)/4096$$

这里的/4 096 除法，因为它是 2 的 N 次方，所以可以用移位操作替代，即往右移位 12bit 就是把某个数除以 4 096。

如图 3.2 所示的代码编译后，又快了 20%。虽然快了不少，还是太慢了一些，20 s 处理一幅图像，我自己都不能接受！但是目前这个式子好像优化到极限

```
Void calc_lum()
{int I;
        for(i=0;i<IMGSIZE;i++)
        {int r,g,b,y;
        r=1224*in[i].r;  g=2404*in[i].g; b=467*in[i].b;
        y=r+g+b;
        y=y>>12;  //这里去掉了除法运算
        out[i]=y;
        }
}
```

图 3.2

了,要想突破音障,只能拆掉活塞发动机,安装涡轮引擎!也就是说,我们必须换一种方式写这个程序了。

仔细端详一下图3.1,RGB的取值似乎有文章可做,RGB的取值永远都大于等于0,小于等于255,我们能不能将D、E、F都预先计算好,然后用查表算法计算呢?把最麻烦的乘法运算预先计算好,使用3个数组分别存放DEF的256种可能的取值,然后预先完成乘法。这样应该可以节约不少时间吧!

使用图3.3所示的代码,预先计算好查表数组,图3.4是真正的查表运算函数。这里没有了乘法运算。

```
Int D[256], E[256], F[256];   //查表数组
Void table_init()
{int I;
        for(i=0;i<256;i++)
        {D[i]=i*1224;  D[i]=D[i]>>12;
        E[i]=i*2404;  E[i]=E[i]>>12;
        F[i]=i*467;  F[i]=F[i]>>12;
        }
}
```

图3.3

```
Void calc_lum()
{int I;
        for(i=0;i<IMGSIZE;i++)
        {int r,g,b,y;
        r=D[in[i].r];  g=E[in[i].g]; b=F[in[i].b];  //查表
        y=r+g+b;
        out[i]=y;
        }
}
```

图3.4

这一次的成绩把我吓出一身冷汗,执行时间居然从30 s一下提高到了2 s!

在 PC 上测试这段代码，眼皮还没眨一下，代码就执行完了。如图 3.5 所示，一下提高 15 倍，爽不爽？

我们已经突破音障！现在让我们踩足油门，向 2 马赫进军！

看了 CPU 的规格书，发现其实很多 embedded sysytem 的 32bitCPU，都至少有 2 个 ALU，能不能让 2 个 ALU 都跑起来？

如图 3.6 所示，2 个 ALU 处理的数据不能有数据依赖，即某个 ALU 的输入条件不能是另一个 ALU 的输出，这样才可以并行。

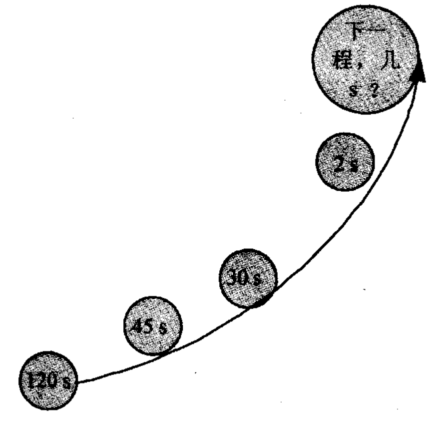


图 3.5

```
Void calc_lum
{int I;
        for(i=0;i<IMGSIZE;i++)
        {int r,g,b,y;
        r=D[in[i].r];  g=E[in[i].g]; b=F[in[i].b];  //查表
        y=r+g+b;
        out[i]=y;
        }
}
```

图 3.6

这一次的成绩是 1 s。再快一些，加足燃料，进军 3 马赫！看看图 3.7 的代码，到这里，似乎已经足够快了，但是反复实验，发现还有办法再快！可以将

```
Void calc_lum()
{int I;
        for(i=0;i<IMGSIZE;i+=2)  //一次并行处理2个数据
        {int r,g,b,y,  r1,g1,b1,y1;
        r=D[in[i].r];  g=E[in[i].g]; b=F[in[i].b];//查表
        y=r+g+b;
        out[i]=y;
        r1=D[in[i+1].r];  g1=E[in[i+1].g]; b1=F[in[i+1].b];  //查表
        y1=r1+g1+b1;
        out[i+1]=y1;
        }
}
```

一次并行处理2组数据
所以这里一次加2

由第二个ALU执行

图 3.7

Int D[256],E[256],F[256]； //查表数组

更改为

Unsigned short D[256],E[256],F[256]； //查表数组

这是因为编译器处理 int 类型和处理 unsigned short 类型的效率不一样（注意：有的编译器可能 unsigned short 比 int 类型更慢）。再把函数 calc_lum 声名成为 inline 函数，将函数声明为 inline，这样编译器就会将其嵌入到母函数中，可以减少 CPU 调用子函数所产生的开销。这 2 个小小的改进带来的效益：速度提升为 0.5 s。现在，如图 3.8 所示，已经达到了客户的要求！

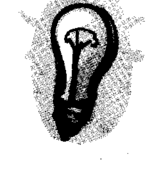
```
Unsigned short D[256], E[256], F[256];  //查表数组
Inline Void calc_lum()
{int I;
        for(i=0;i<IMGSIZE;i+=2)  //一次并行处理2个数据
        {int r,g,b,y,  r1,g1,b1,y1;
        r=D[in[i].r];  g=E[in[i].g]; b=F[in[i].b];  //查表
        y=r+g+b;
        out[i]=y;
        r1=D[in[i+1].r];  g1=E[in[i+1].g]; b1=F[in[i+1].b];  //查表
        y1=r1+g1+b1;
        out[i+1]=y1;
        }
}
```

图 3.8

其实，如果再加上以下措施，应该还可以更快：

● 把查表的数据放置在 CPU 的高速数据 CACHE 里面。

● 把函数 calc_lum()用汇编语言来写。



其实,CPU 的潜力很大,

● 不要抱怨你的 CPU,记住一句话:"只要功率足够,砖头都能飞!"。

● 同样的需求,写法不一样,速度可以从 120 s 变化为 0.5 s,说明 CPU 的潜能很大!看你如何去挖掘。

● 我想:要是 Microsoft 的工程师都像我这样,反复优化代码的运行速度,那大概就可以用 486 跑 Windows XP 了!

4

设计的中庸之道

中国的知识分子,大概都听说过"中庸之道"。对于"中庸之道"有很多种说法。我常听人一提到"中庸之道",第一反应就是"不冒尖、不落伍"之类的随大流的思想。他们认为走在队伍的中间就是"中庸之道"。个人认为,这其实是严重错误理解了"中庸之道",至少这也是很片面的理解。

我所理解的中庸之道的内涵大致有以下几点:

- 以中道行事,不走极端。
- 执两用中,实行稳健的主张。
- 符合社会人生常道,做到合情合理。
- 宽容包纳,和而不同。

以下段落分别叙述这4点。

4.1 以中道行事,不走极端

孔子说:"过犹不及",孔子不赞成偏激行为,不论是左偏还是右偏,他主张中立而不倚。那么中道的标准是什么?就是仁义之道。孔子说:"唯仁人能好人,能恶人","行义以达其道","君子义以为上"。仁义之道就是儒家的人道主义和社会公正原则。有人把中庸之道理解成不讲原则的折衷主义,其实是不对的。不讲原则,四面讨好,自私伪善,孔子称之为"乡原"。"乡原,德之贼也",孔子对之深恶痛绝。所谓不偏,是不偏离人道精神和社会公正,这是一个基本坐标。

先用医学方面的例子,来说明这个"以中道行事"。以下列出一些我所知道的医学方面的走极端的不好:

- 平时的膳食中,如果荤吃多了,容易形成心血管疾病。

- 如果谁只吃素，虽然比偏向荤食好处多很多，但是他同样也是结石病的高发人群。因为素食中经常含有大量的草酸。
- 如果不经常运动，容易引起代谢水平降低，引发形形色色、稀奇古怪的毛病。
- 如果太经常运动，比如运动员和重体力劳动者，极少有长寿的，而且猝死的概率明显高过一般人群。
- 营养过剩，会导致心血管疾病、糖尿病、痛风等疾病。
- 营养不良，会导致低血糖等疾病。肺结核和麻风病也多发于营养不良的人群。
- 以站立姿势工作的人，比如营业员，是痔疮，静脉曲张的高发人群。
- 以坐姿工作的人，比如 IT 工作者、公务员等，是颈椎病、腰椎病的高发人群。

综合以上几点，不难得出以下结论：

- 饮食不要偏荤，也不要偏素，而要合理搭配。
- 不能长期高强度运动，更不能从不运动。
- 不要营养过剩，也不要营养不良。
- 不要长时间站着，也不要长时间坐着。

这难道不是解释“以中道行事，不走极端”的最好例子吗？

图 4.1 是一个电话机通话部分的框图。众所周知电话线只有 2 根线，声音的收和发都在其上。电话机的电路做得不好的话，就会出现自己在麦克风里面讲的话，在自己的听筒里面放大后重放出来。这样轻则给使用者的感觉不好，重则扬声器的声音传播到麦克风里面，形成正反馈而啸叫。所以电话机的国家标

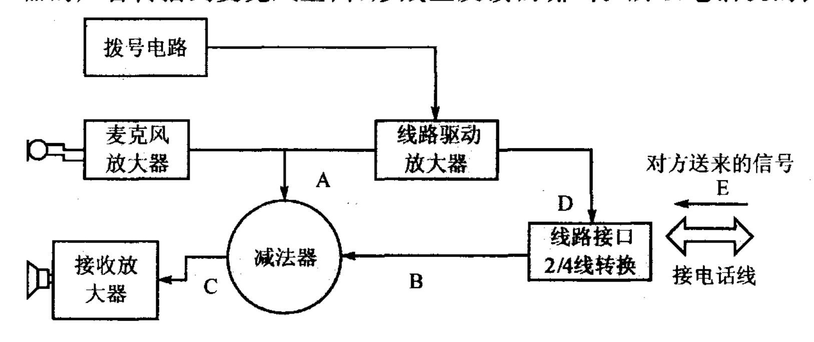


图 4.1

准里面有一个关于消侧音的要求。消侧音的大致意思就是说，电话机的麦克风发送的信号，在受话器中重放出来，其电平要低过多少分贝。

这个消侧音电路是非常难调试的电路。图中信号 A 是经过放大的麦克风音频信号，D 是和 A 相位相同的信号，所以 B 信号也包含了一部分 D，还有一部分对方送过来的语音信号 E。以上各个信号各有不同的幅度，分别用小写的 a、b、c、d、e 来表示信号 A、B、C、D、E 的幅度。

用数学公式描述这个问题，就是

$$B*b=D*d+E*e$$

$$C*c=K*(A*a)-B*b=K*(A*a)-D*d-E*e$$

$$C*c=K*a*A-D*d-E*e$$

而如果要在信号 C 中尽可能的降低麦克风的信号，只要使以上算式中的 $K*a=d$ 就可以完全消除侧音，若等式不满足，麦克风的信号就会在信号 C 和扬声器中出现。而减法器中的相减比例是通过一个电阻调节的。本来调节侧音用电位器最好，但是电位器成本高，而且可靠性也低一些，所以电话机中都是在设计阶段调节侧音，量产的时候就不再一一调节了。但是为了确保产品在出厂的时候，不超越国家标准的限制，所以工厂的内部控制标准比国家标准更严格（如图 4.2 所示）。而为了在生产中不超越工厂的内部控制标准，则在设计的时候，就必须非常精确的把侧音调节到最小值。这样在量产的时候，无论是由于元器件的离散

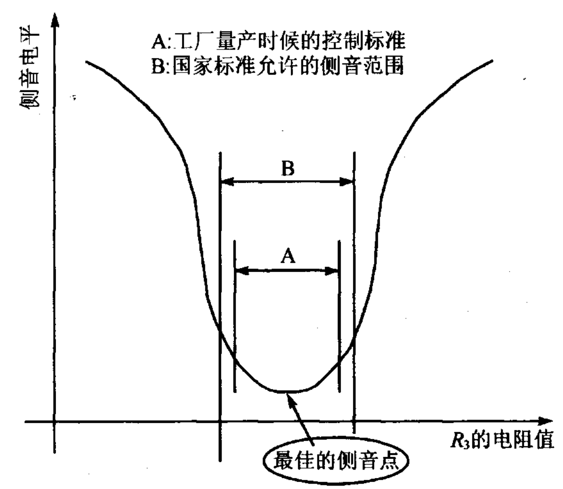
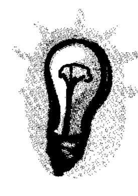


图 4.2

性造成的参数漂移还是由于温湿度变化造成的参数漂移，都可以把量产产品的侧音参数控制在范围 A 之内。

图 4.3 和图 4.4 是电话机集成电路 TEA1062 消侧音的信号回路，它基本符合图中的信号流程。精确调节侧音到最小点，就是不断地修正 R_3 阻值，达到最佳点。R_3 太大、太小都不会得到好的侧音值。

图 4.3



提示：

- 常在河边走，一定会湿鞋。远离河边，你就会很安全了。
- 要把硬件设计在安全状态，就要把硬件调节到最安全的那个点上。

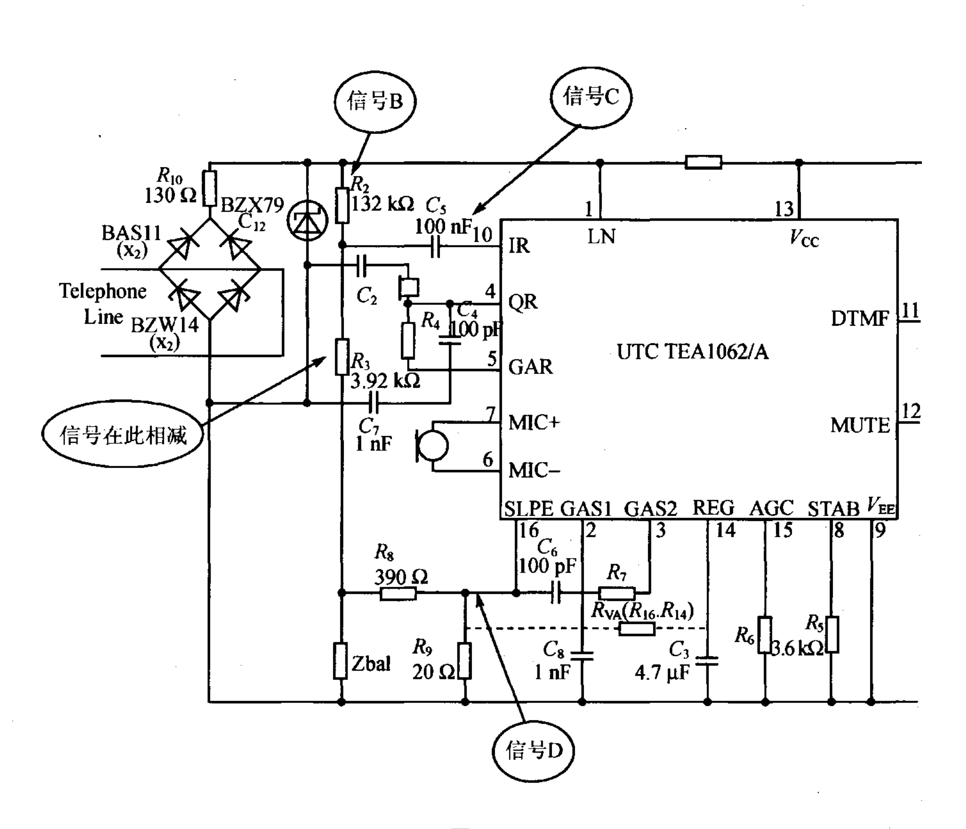


图 4.4

4.2 执两用中,实行稳健的主张

《论语》曰:"允执其中","叩其两端",《中庸》曰:"执其两端,用其中于民。"这"两端"并非仅指两个极端,而是指各种不同的意见,特别是正面和反面的意见。执两用中的原则,要求主事者要多听赞成的和批评的意见,然后全面考察,采取比较客观和稳妥的主张加以实行。从思维方法上说,这是考察事物矛盾的两个对立面而后加以统一;从认识论上说是集思广益尔后作出科学的判断;从实践行为上说是照顾各群体的实际利益而有所妥协,在异中求同,以便达成一致,实现合作。所以中庸之道尽管不是折衷主义,但有折衷的成分,是一种积极的调和论。

在二战后期,盟军经过几年的修整和战略储备,准备反攻欧洲大陆,在欧洲

大陆的南面开辟第二战场。最后决定选择诺曼底为登陆地点，在选择登陆时间的问题上，海军司令和陆军司令争执起来。陆军司令说："我们要在涨潮的时候登陆，因为那样的话，我们的士兵可以更加靠近德国人的阵地，可以节约几百米路程，这几百米路程就有可能造成很多人的伤亡。后面也要有离海岸线近一些的军舰的火力掩护，以压制住德军的火力"；而海军司令则认为："我们大型舰船不能离陆地太近，否则在德军炮火的射程之内。即使用大吨位的舰船护航，其火炮的射程也不及陆基火炮的射程。而舰船的机动性和隐蔽性都不是非常好，目标也大，与居高临下的陆基火炮对决，舰船可能吃亏大一些。"最后二者争执不下，达成一个妥协：潮水在中间状态的时候进行抢滩登陆。这样二者的利益都得到了平衡。而后来的战局证明了这个妥协是对的。

因为我喜欢摄影，所以经常有人来和我讨论关于摄影的话题，去年有个有钱人来问我，数码相机应该怎么买。我首先问他："你买数码相机，照片质量和照相机的轻便性，你优先选择哪一个？"他底气十足地说："钱不是问题，我都要！"。我只好婉转地告诉他，二者不可兼得，这个问题不是用金钱就可以把二者全部一次性解决的。因为照片质量好的照相机，其光学系统的性能都相当优越，而这个优越的代价就是大尺寸：大尺寸的镜头、大尺寸的镜片、大功率的闪光灯、大容量的电池和大尺寸的光学成像 IC。而注重轻便性的照相机，它必然要压缩光学系统的机械尺寸，随之而来的就是照片画质的下降。最后在我的建议下，他就折中选择了一个尺寸中等，画质也中等的机型。

上次看 CCTV 的《每周质量报道》，里面谈及汽车刹车片的品质问题。其中特别提及了国家标准对汽车刹车片摩擦系数的要求。按照一般的思维，既然是刹车片，自然是摩擦系数越大越好。可是国家标准不仅规定了它的下限，偏偏还规定了汽车刹车片的摩擦系数的上限，理由是：因为摩擦系数太大的话，在急刹车的时候，刹车系统可能会抱死轮胎而造成汽车侧滑，最少也会造成轮胎滑动而刹车距离增大。所以国家标准这样定，其实也是兼顾了二者的利弊。

不知道你有没有戴过士兵的钢盔，反正我是戴过。戴上钢盔的第一感觉就是："好重！不能做得轻便一些吗？"。后来我才知道，即使钢盔这样设计，虽然可以挡住很多流弹和弹片的冲击，但是仍然挡不住 300m 内射来的步枪子弹，甚至挡不住 50 m 之内的手枪子弹。如果钢盔要做到绝对保险，估计那个重量没有谁可以承受。除非将来有什么抗冲击的新材料出现，否则钢盔就只能在安全和重量之间取平衡。这样平衡的结果就是：既有一定的轻便性，又有一定的安

全性；同时它既不是绝对的轻便，也不是绝对的安全。

图 4.5 是一个 AMD 速龙处理器规格书的节选。大家可以看到其 CPU 的峰值功率有 89 W 之多。那我们也许就会问了：AMD 的工程师为什么不降低那个 1.50 V 的电压，这样功率不就降下来了吗？想法是很好，可是因为这个 CPU 工作频率极高，电流也比通常的 IC 高 10～1000 倍。所以对噪声会比较敏感，虽然调低电压可能会降低一点功率，可是同时降低的还有可靠性。因为噪声不会随着系统的电压下降而下降，IC 的工作电压低到一定程度，电路噪声对稳定性所造成的破坏力就越大。这就是为什么 CPU 超频的时候，一般都要把 CPU 核心电压往上微调一点的道理。如果牺牲节能的性能，只追求可靠性，CPU 的可靠性会不会成比例提高呢？当然不会，因为道理很简单：一个高发热的系统，各种材料都会加速老化、电路板的焊点也会加速老化，散热系统的造价也会提高。所以这样设计其实也是在耗电和可靠性之间的一个无奈的平衡。

AMD

30430 Rev. 3.43 October 2004 *AMD Athlon™ 64 Processor Power and Thermal Data Sheet*

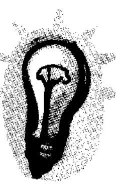
Parameter/OPN	**ADA3000AEP4AR**	**ADA3200AEP5AR**	**ADA3400AEP5AR**
Model Number	3000+	3200+	3400+
CPUID 8000_0001h EAX [31:0][7]	00000F4Ah	00000F4Ah	00000F4Ah
FID/VID Status MaxVID Field[8]	00h	00h	00h
FID/VID Status MaxFID Field[8]	0Ch	0Ch	0Eh
FID/VID Status StartVID Field[8]	02h	02h	02h
FID/VID Status StartFID Field[8]	0Ch	0Ch	0Eh
L2 Cache Size	512 KB	1 MB	1MB
T_{CASE} Max[11]	70°C	70°C	70°C
$T_{CONTROL}$ Max[12]	70°C	70°C	70°C
Max P-State	2000 MHz	2000 MHz	2200MHz
VID_VDD	1.50 V	1.50 V	1.50 V
IDD Max	57.8 A	57.8 A	57.8 A
Thermal Design Power[1]	89 W	89 W	89 W
Thermal Resistance (case-to-ambient)[9]	0.31°C/W	0.31°C/W	0.31°C/W
Intermediate P-State #1	1800 MHz	1800 MHz	2000 MHz
VID_VDD	1.40 V	1.40 V	1.40 V
IDD Max	45.6 A	[illegible]	47.7 A
Thermal Design Power[1]	66 W	[illegible]	69 W

图 4.5

刚参加工作的时候，经常设计交换机的接口产品。那时候，语音通信设备很贵，一个 2 M 的光端机就要 1～2 万元。以现在的眼光看，一个传送速率 2 Mbps 的设备要 1～2 万，实在是太贵了，可是当时就是这个价。懂一点电信传输知识的人都知道，这样一个 2 Mbps 的数据链路，其实只能同时传递 32 个语

音话路，每个话路是 64 Kbps 的速率(8 kHz 采样，每次采样 8 bit)。当时我就有疑问，要是把采样率降下来，带宽不就小了吗？那每个光端机传递的话路不是就可以多一些吗？但是后来了解了电信传输原理才知道，之所以这样选择也是有理由的：

- 当时能定义数据链路格式的，就是世界上的少数几个公司，其他的制造商都是后面的跟屁虫。这里面也包括我们。
- 当时没有硬件的数据压缩芯片，所以所有的数字信号都是不压缩传送的。
- 如果把语音的采样率和采样位宽下降，当然可以达到我所说的“每个光端机的传递的话路不是就可以多一些”的目的，但是代价是语音信号质量的下降。
- 如果拉高语音信号的采样率和采样位宽，虽然语音信号质量上升了，但是带宽变多了，这样就要拉高整个电信网络的造价。
- 所以在带宽和语音质量的互相矛盾下，他们最终选定了 8 kHz 采样，每次采样 8bit 的形式，并成为世界程控交换机的标准。
- 由于总是要兼容旧设备，现在无论是有线的交换机还是移动通信的交换机，它们的数据交换格式都是 8 kHz 采样，每次采样 8 bit 的形式。



提示：

- 任何事物都有两面性，解决问题的时候，既要考虑到它的正面作用，也要考虑其反面作用。
- 当二者不可兼得的时候，就必须进行折中处理。

4.3 符合社会人生常道，做到合情合理

“不偏之谓中，不易之谓庸。中者，天下之正道；庸者，天下之定理。”中庸即是用中之常道，它的最大特色是人情与道理的协调，符合多数人的健康正常生活需要。孟子说：“仲尼不为已甚者”，所谓“甚”就是不合于情理。人性之中有三大要素：情欲、德性和理智。德性是调节情欲与理智的杠杆。情欲强而理智弱则流于放荡，理智强而情欲弱则失于冷酷。一个健康的人性，应是情与理的平衡，他所做的事情便会合情合理，为多数人所接受，这是一种理性精神。

当然人生常道并非一成不变，因为生活在不断变化，所以孟子说孔子是“圣

之时者也”。《爱中庸》讲“君子而时中”。朱熹说“盖中无定体，随时而在，是乃平常之理也”。中庸之道接近生活，又是健康的，所以它既是平常的，又不与陋习同流合污，所谓“和而不流”者是也。

这个道理虽然简单，但是要做到的话，却需要对所要解决的事务，有清醒而深刻的认识。否则就很难做到“合情合理”。

我举一个设计不合理的例子，是一个 EDN 论坛上的帖子，图 4.6 是其所配的插图。

这是他的提问：

各位前辈们好：

新进公司，头头安排做一个 12 V 铅酸蓄电池充电电路，要求电路简单，能防过充，不知如何下手，网上有用 UC3906 做的，但芯片价格偏贵，这是另一个电路，给 2 节 12 V 电池充电，请教各位前辈们，可以指导我一下此图电路原理吗？如果要改成只供一节电池充电，如何下手？不胜感激！！！

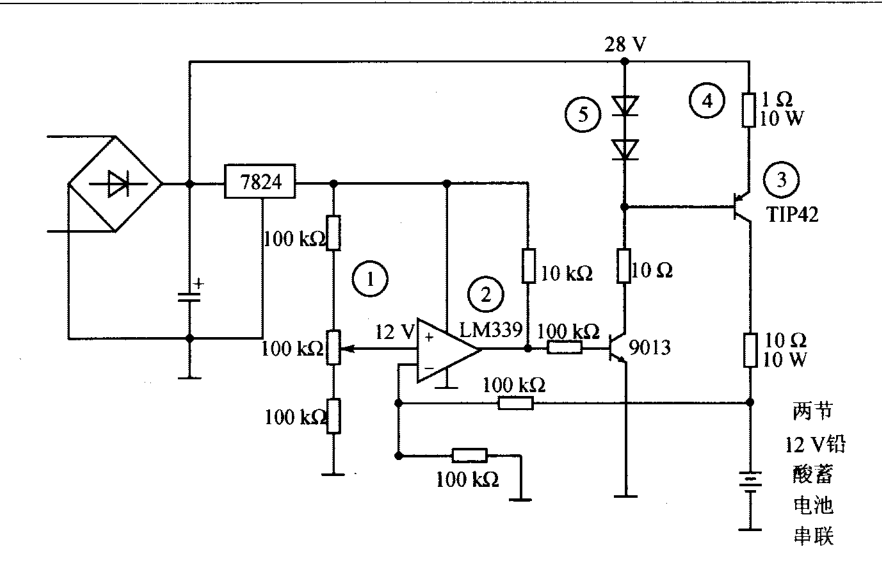


图 4.6

图 4.6 我用 1 分钟就看出了其中的 N 个错误（如表 4.1 所列），估计这是一个菜鸟画的。这个设计十分不合理，既没有考虑电路量产时的便利性，也没有考虑电池是否能长寿，更没有考虑充电器本身是否节能。

表 4.1

<table>
<tr><th>标 号</th><th>描 述</th></tr>
<tr><td>①</td><td>首先一个错误，这里不应该使用电位器。电位器在生产的时候多一个调整的工序，像电池充电那种需要精确控制充电电压的电路，如果稍微调偏一点，要么电池不能充饱，要么电池就会过充。而且电位器很不耐酸碱环境，长期在酸碱度大的空气中（铅酸蓄电池工作的环境经常就是强酸性环境），电位器老化速度快 10 倍都有可能。

这个电路的设计者，是想利用 7824 的稳定电压，取一半出来，作为稳定的参考源，以完成恒压控制。但是他可能不知道，稳压三端的电压稳定度是不需要那么高的，一般在额定值的上下 3%～5%都可以。图 4.7 是 7824 稳压 IC 的规格书，该 IC 厂家说了：23～25 V 的输出都是正常的。所以当采购一个大批量的时候，很可能会有一部分 IC 的电压就会落在 23 V 或者 25 V。这样经过电位器分压后，就会对应的少了 0.5 V 或者多了 0.5 V 的参考电压。而作为铅酸蓄电池用恒压充电的充电方式来说，少 0.5 V 电池就充不饱，多 0.5 V 电池就过充

7824 电特性(除非另有规定:0≤T_J≤+125 ℃, V_I=33 V, I_O=500 mA, C_I=0.33 μF, C_O=0.1 μF)
<table><tr><th>参数名称</th><th>符号</th><th>测试条件</th><th>最小值</th><th>典型值</th><th>最大值</th><th>单位</th></tr><tr><td rowspan="2">输出电压</td><td rowspan="2">V_O</td><td>T_J=25 ℃</td><td>23</td><td>24</td><td>25</td><td rowspan="2">V</td></tr><tr><td>5 mA≤I_O≤1000 mA, 27 V≤V_I≤38 V</td><td>22.8</td><td>24</td><td>25.2</td></tr></table>
图 4.7</td></tr>
<tr><td>②</td><td>这里不应该使用 LM339，LM339 是电压比较器，而不是运算放大器。这个电路一看就知道，整个电路处于深度负反馈的线性状态，这时候，用运算放大器才是上上之选。所以这个电路，我觉得用 LM358、TL084 或者 LM324 都可以，只是别去用比较器。因为比较器的输出非 0 即 1，输出中间电压，对它来说不适合</td></tr>
<tr><td>③</td><td>这个三极管在充电的时候，处于线性状态，发热会很严重。因为在充电的时候这个管子会有 3～6 V 电压降，假设充电电流 1 A，那这个三极管就浪费了 3～6 W 的能源。所以从节能的角度说，这样的线性充电电路架构，是不合理的。发热大的电路，电路的可靠性一般也会低一些。因为长期的发热，容易造成器件和电路板焊点加速老化</td></tr>
<tr><td>④</td><td>这里串联一个电阻，电路的设计者大概是想构成一个射极跟随器，以稳定电路。但是这个电阻有点多余，而且同样浪费能源。因为这个电路本身是一个深度负反馈的线性状态，有很强的自我调节和自我控制能力，所以这个电阻，非常多余</td></tr>
<tr><td>⑤</td><td>想不出来加这 2 个二极管的意图，在一个深度负反馈的线性环路里面，加这样的二极管是多余的</td></tr>
</table>

那如何改进呢？如果还沿用这个电路的线性恒压架构进行改进，且按照论坛帖主所说的“改成只供一节 12 V 电池充电”，改进电路图如图 4.8 所示。

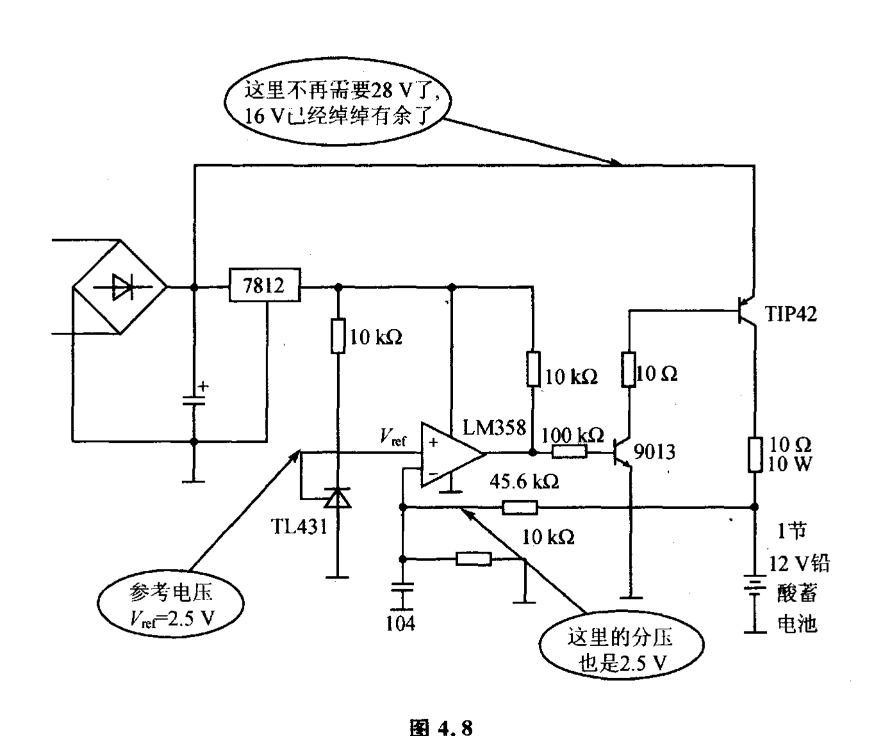


图 4.8

我改进了以下几个地方：

- 使用 TL431 作基准源，电路的稳定性大幅度提高。图 4.9 和图 4.10 是节选自 TL431 规格书的片段，是关于其稳定性的最好说明。
- 取消了电位器，减少了加工工序。也降低了因为电位器失效导致整个电路板失效的可能。
- 主放大器更改成了更适合线性工作，且非常便宜的 LM358。
- 运算放大器的“－”输入端对地接了一个 104 电容，这样可以滤掉由电池方向而来的噪声。
- 参考电压由 12 V 下降到了 2.5 V。
- 恒压充电的终止电压，设计成 13.9 V，而不是 12 V。这样更符合铅酸电池的特性。
- 7824 更换成了更容易采购、更便宜的 7812。
- 7812 的输入电压从 28 V 降低到 16～18 V，这样可以降低 7812 输入端前面电路的造价。因为电压高了，所用到的器件基本都会贵一些。

这样更改后，电路应该能可靠工作。它的稳定性应该也比原先好得多，但是这样就合理了吗？肯定不是。起码 TIP42 的发热还是很大，不够绿色、不够环

这里最多只有正负1%的偏差

ELECTRICAL CHARACTERISTICS at 25 ℃ I_k @ 10 mA $V_k=V_{ref}$ unless otherwise specified.

Parameter	Symbol	Test Conditions	Test Circuit	TL431			Unit
				Min	Typ	Max	
Reference Voltage	V_{REF}	T_A=25 ℃ Over Temp.	1 1	2.470 2.449	2.495	2.520 2.541	V

图 4.9

FEATURES

- **Trimmed Bandgap to <u>1%</u>**
- **Wide Operating Current..........................1mA to 150mA**
- **Extended Temperature Range...................0°C to 105°C**
- Low Temperature Coefficient ....30 ppm/°C
- **Offered in TO-92, SOIC, SOT-89, SOT-23-5**
- **<u>Improved</u>** Replacement in Performance for TL431
- **Low Cost Solution**

极好的温度稳定性

图 4.10

保。还有，恒压充电本身也不是很合理，现在大部分使用的充电方法都是二阶段法。二阶段法采用恒流和恒压相结合的快速充电方法，如图 4.11 所示。首先，以恒电流充电至预定的电压值，然后，改为恒电压完成剩余的充电。

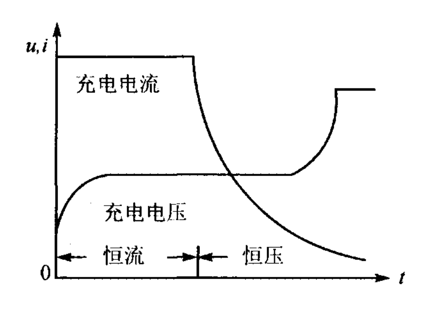
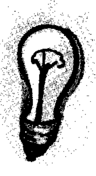


图 4.11　二阶段法曲线

所以，要把这个充电器设计得完美，应该抛弃 TIP42，改用开关电源，降低充电器本身对电能的损耗，同时为了实现二阶段法充电，加入一个小小的 CPU 也是必须的。由于这个已经超出了“中庸之道”的讨论范围，这里就不叙述了。

以前有人设计过一个电池供电的手持设备，在实验室和工厂测试都是好的，但是样品提交给用户后却反映容易死机，而且一般都是提交后 3～4 个月才有死机的报告。为此他们罗列了好几种可能：元器件老化、焊接和组装的工艺不良、电池电压下降等。其中为了测试是否是电池电压下降诱发的故障，就用稳压电源人工调低电压，果真发现电压下降到 80％的时候，死机的概率大大上升了。这说明设计者在设计的时候，没有考虑电压的适应性，或者对电池的特性不熟悉。电压跌落的时候，也没有任何指示。用户在这个时候，很容易认为是机器发

生了硬故障。如果这个时候有相应的低电压提示，用户的感觉就会好很多。所以这个也是其设计不够人性化之处。



说了这么多，归根结底是要大家知道做产品应该方方面面都要做到合情合理。这是很考验人的经验和智力的，也很考验您是不是有足够的耐心去把每一个细节做好。

4.4 宽容包纳，和而不同

中庸之道与“和为贵”的思想相结合，便是中和之道。《中庸》说：“喜怒哀乐之未发，谓之中；发而皆中节，谓之和。”这是从人性论角度讲的。人的本性是淳朴的，并无偏失；而后天的性情则有中节与不中节之分。所谓中节，就是符合社会行为的一般规则，能与社会人群和谐相处。淳朴人性是天下之大本，和谐原则是天下之大道，离开中和之道，便会人性浇漓，天下大乱。实行中和之道的关键便是使人的行为符合忠恕之道，己欲立而立人，己所不欲勿施于人，做到互爱互信，互尊互谅，人得其所，事得其宜，则天下太平。孔子提出的“和而不同”的原则最能体现中和之道的精神，它是正确处理人际关系的黄金规则，具有最普遍的意义。

一个使用 ARM7 构建的 ucLinux 系统，我们在为其开发应用软件的时候，曾经碰到一个匪夷所思的问题：

- 该系统内存有 64M 字节之多。
- ucLinux 操作系统本身占用了一部分内存，但是还给应用程序至少留下了 32M 的可用内存。
- 我们的程序是一个流媒体解码的程序，经常需要分配大块的内存。有时候，一次可以申请大于 1M，而且经常多次。
- 我们的代码中，申请内存和分配内存非常频繁。
- 这样，我们的麻烦来了：我们的软件经常会崩溃，监视串口的输出信息，发现崩溃的原因基本都是因为内存分配的原因。先是申请某个内存，然后因为内存不足，而导致系统崩溃。

后来我们仔细审查了代码，虽然申请内存和分配内存非常频繁，但是经过计算，剩下的 32M 内存应该足够我们做这样的操作。为此我们困惑了很久。后来查阅文献才知道：这是 ucLinux 操作系统固有的毛病，在 ucLinux 操作系统释

放大块的 memory，然后应用程序再马上申请一个同样大小的 memory 就有可能不能马上申请得到。

为了证实这种说法，我们抛开这个已经写了 20,000 多行的工程，建立一个很小的工程，这个工程只有 200 行的代码，它干的事情只有一个：不停地分配和释放大块的内存。这个程序跑起来后，果真看见了相同的程序崩溃。

问题确认了，但是怎么处理却想了很久。因为这是操作系统本身固有的毛病，最直接的想法，当然是：操作系统有问题就要去解决操作系统的问题，而且这个操作系统是开源的，修改它是有可能的。但是问题在于：这个操作系统的源代码有百万行之多，别说修改它，就是找到要修改的地方都要耗费至少 10～20 天的时间。也就是说，这个解决办法理论上可行，但是执行的成本太高。

为此，又想了很久。如果我们擅自在别人开发的操作系统核心层上动刀子，即使能解决这个问题，也极有可能会出现未知的新问题。最后我们认为既然问题发生在频繁申请内存和分配内存上，那如果不要那么频繁申请内存和分配内存，问题是不是可以解决呢？于是做了如下的改进：

- 程序一启动，马上把需要的大块的内存申请下来。
- 申请完后，在程序中把这些大块的内存当作固定内存来操作。不再释放。
- 需要使用大块内存的地方，都使用这些预先申请的内存。

这样改进后，就解决了以上的问题。从而避免了对操作系统本身动手术。这个问题的解决就体现了软件设计时的“宽容包纳，和而不同”。这样改进后，操作系统原来有什么毛病，现在还有什么毛病，但是只要通过改进应用程序绕开这个问题，就能解决问题。

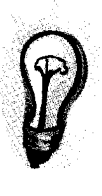
本书“搞定故障”一章中的“处理按键背景灯老是会随着音乐亮度明显变化”的案例，其实也用的是这个原理。

我记得以前修理自家的洗衣机，打开机器后背来修理，要拧开 8 个螺丝，如果细心观察的话，可能会发现主机上的螺丝孔是圆形的，刚好够一个螺丝进去。而后盖的螺丝孔却统统被加工成椭圆的，这样洗衣机的后盖板在安装的时候，经常可以上下调整 1～2 mm 的距离。这样的设计就是考虑到洗衣机主机和后盖板量产时的加工公差，后盖板这样设计，洗衣机机身螺丝孔少许的定位公差不会影响螺丝的紧固。这样的设计，其实也是在一定程度上允许洗衣机机身螺丝孔的定位误差，增加对误差的宽容度来达到量产时的稳定。





再举一个医学的例子。众所周知一个人如果肾衰竭后需要移植肾脏，而移植肾脏首先需要配型。所谓配型就是挑选那些排斥反应发生概率比较低的个体。但是即使配型成功，其发生排斥反应的概率还是要比原装肾脏要高无数个数量级，只是这种排斥反应已经低到可以用人工手段控制而已。正因为无法找到和原装的肾脏一样的没有排斥反应的配件，就不得不容忍还有一点排斥反应的肾脏。所以肾脏移植成功后，病人就必须终生服用免疫抑制剂，将人体的免疫水平压到正常水平以下，以减少排斥反应发生的概率。同时病人还不能吃可以提高免疫力的食品，比如香菇和人参等。虽然人为降低免疫力后会给患者带来很多麻烦，但是这也是没有办法的事情，因为找不到完美的配件，只能通过这些手段来容忍现有配件的缺点，只有这样它们才可能比较好地磨合在一起。所以，这同样也是一种"宽容包纳，和而不同"。



提示：

- 人与人的交往中，每个人都有缺点。我们不能要求别人都没有缺点，适当的容忍别人的缺点，也是保持和谐的主要手段。
- 电路中的各个模块也多多少少会有对外的干扰。要求别的部件都十全十美是比较困难的，所以有时候，把电路的抗不良环境的能力设计得强一些，也同样是把电路做好的一个窍门。

5

搞定故障

做研发，查找故障是家常便饭。这个查找故障可以是修理别人设计的成品整机的时候的查找故障，但多数是研发过程中的查找故障，检查一个硬件小模块的故障，检查一个电路板的故障或是检查一段软件代码的故障。

查找故障，就其思维方式而言，和医生看病、警察探案没有本质的不同，差别只是在各自的专业知识领域。本文着重写三者共性的东西，即查找故障的思维方式。

5.1 望、闻、问、切，收集故障信息

无论是医生看病、警察探案还是诊断一个硬件或者软件的故障，首先都需要收集信息。收集信息最经典的概括就是中医所说的“望、闻、问、切”。其实不仅中医讲究“望、闻、问、切”，西医也是一样的，西医医生也同样会根据人体的气色、精神状态来大致判断一个人的健康程度；用血压计、CT、B 超、X 光来做主动性的信息收集工作，有了这些信息，开处方才不会出差错。

5.1.1 望和闻

望和闻是一种被动的信息采集过程。通过望和闻来知晓别人是什么年龄、什么性别、大致什么地方有疾病、疾病的严重程度等。甚至一些有经验的医生可以估计出患者大致是什么职业、平时的生活习惯和饮食习惯等信息。

比如电池和连接器是电路的故障高发元件，拿到故障机器以后，完全可以依据电池的成色来初步判断电池是否老化；依据连接器的做工是否精致、触点是否有氧化等眼睛可以看见的地方来判断连接器是否可能会有故障。对于那些因为过流和过压烧坏的电路板，也可以通过眼睛看、鼻子闻来确定什么地方曾经被烧灼或者击穿。也可以询问故障机器的使用者，看看故障发生的时候，用户有哪些

和设计者不一致的操作习惯。这些信息都会对故障查找很有帮助。

5.1.2 问和切

问和切是一种主动的信息采集过程，比如看病的时候医生会问：最近胃口好不好？有没有咳嗽？有没有发烧之类的。通过这些问题，进一步缩小包围圈。如果单单病人的口述还不够，那就要“切”了。“切”可以是切脉、可以是测量血压、CT、B超、X光等技术手段，获得十分详细的信息。有了这些信息，医生开出处方，就十拿九稳了。

同样的，在处理故障的时候，也经常要主动性的探查，比如用示波器和万用表等设备，对电路做检测。这样就可以得到一些靠人体的五官得不到的信息。比如电压、电流、波形的形状和频率等信息。这些信息对处理故障都非常有用。

5.1.3 诱

这个是我在“望、闻、问、切”之外自己加的，就是“引蛇出洞”的意思。其实“诱”可以归类在“切”里面，都是主动性的探查。但是不同的是上面所说的“切”是不破坏被测量物体的内在属性和特征的前提下，去主动性的探查。而“诱”带有一定的引导性，在一定程度上会影响被测量物体的内在属性和特征，通过这种主动性的影响，来达到发现问题的目的。其实警察在办案的时候，也经常会用这一计策。在圈定了几个嫌疑对象而不能最终确定的时候，做一个局让罪犯自己跳出来，这样的例子在电视上的法制节目中有很多。

电视上看见石油工人在戈壁滩上找油、找天然气，需要绘制地下岩石的断面图，但是又不可能把那么大的一片大地全部剖开。于是他们就隔 1～2 km 挖一个深坑，一共几十个坑，组成网格状的深坑群。在每个坑里面都放上几吨炸药。然后再在安全的地方分散安装上一大把传感器。把那些炸药逐个引爆后，制造人工的小型地震，用计算机处理各个传感器测量到的回波数据，建立地层的影像。这个方法就是通过“诱”的方法，对地层进行小规模的破坏，用几十吨炸药的小代价，一下可以测绘出几百平方公里范围内的地层影像。

对于一些发生概率不高的故障，或者无法预测发生的时候，故障的后果有多少的严重程度，都可以用“诱”。创造条件诱发故障，这样可以节约时间。对医学诊断而言，除非万不得已，否则将疾病进行诱发和放大，都是不道德的。所以这一方法在医学中较少使用。但是现在研究的是如何搞定机器的硬件和软件，完全可以为了节约时间而制造条件让故障出现，即使付出一定的代价，有时候也是

合算的。

以前有人设计过一个电池供电的手持设备，在实验室和工厂测试都是好的，但是样品提交给用户后却反映容易死机，而且一般都是提交后3～4个月才有死机的报告。为此工程师罗列了好几种可能：元器件老化、焊接和组装的工艺不良、电池电压下降等。其中为了测试是否是电池电压下降诱发的故障，就用稳压电源人工调低电压，果真发现电压下降到80%的时候，死机的概率大大上升了。这说明设计者在设计的时候，没有考虑电压的适应性。这就是创造条件，诱发低概率故障的案例。

以前我经常修理电话机，经常会有人在阴雨天后拿电话机来维修。用户陈述的故障现象基本都是"电话呼入后，响一声就没了。"刚开始并不知道这种故障和潮湿天气有关，而且用户把机器拿来的时候却经常在艳阳天，机器到我手上，故障已经不存在了。所以总是认为顾客的沟通有问题。但是经过几次反复，才开始怀疑和潮湿天气有关。为了确定是否是潮湿引起的，将电话机电路板和一块沾满了水的卫生纸，一起放在一个大塑料袋中，二者不直接接触，然后扎紧封口。24小时后，再测试这些电路板，无一例外的都出现了"电话呼入后，响一声就没了"的故障，原来电话机厂商为了降低成本，一般是使用廉价的纸基板做电路板的板材。纸基板的最大毛病就是不耐潮湿环境，是潮湿导致了这些纸基板绝缘性能下降。

图5.1是其中一个比较典型的电话机的电原理图(局部)，图中A点是电话线的电压，挂机的时候有DC24 V或者DC48 V，加上振铃的时候，最大可能会有(90+48) V的峰值(振铃电压国标为AC 90 V)。这里V_2和V_3控制着电话机是否开始通话。它导通，电话机进入低阻状态，交换机会认为电话机摘机，开始通话了；否则电话线呈现高阻状态，交换机会认为还没有摘机。而这两个三极管又被B点的电压所控制，B点为高电平的话，V_2和V_3就会全部导通。

而出现故障的机器，在电路板上又大多是A、B两点靠得比较近。有的甚至是平行走线，这样碰见纸基板且空气潮湿的情况下，相当于在A、B两点之间跨了一个1 MΩ～十几MΩ的电阻，所以一旦A点电压非常高的时候(振铃的时候，最大可能会有(90+48) V的峰值)就会导致V_4导通。V_4导通后又导致V_2和V_3导通。V_2和V_3导通后，由于电话线呈现低阻抗，所以A点的电压又往下掉，导致V_4、V_2和V_3全部截止，电话重新挂断。所以就会出现"电话呼入后，响一声就没了"。为了确认以上的分析，在电路板非常干燥的时候，在A、B两点之间人为接入一个5.1 MΩ的电阻，也会出现一模一样的故障。这样，就对故障做

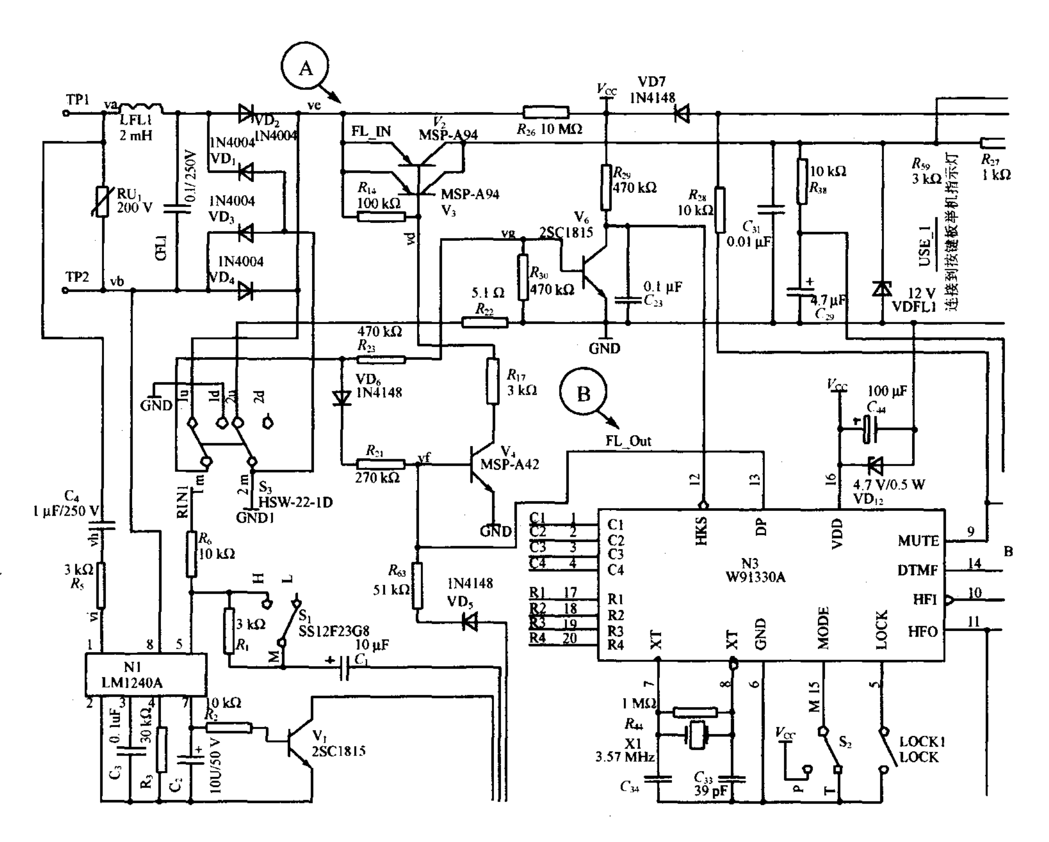


图 5.1

了第二次诱发，进一步确认了病因。

确诊原因后，处理起来就简单多了，解决方法如下：

- 在电路板上将线条 A 的全部铜箔去掉，只剩下焊点。
- 然后用塑料皮的焊线将线条 A 上的所有焊点，重新连接起来。
- 这样 A、B 两点之间的绝缘电阻就会大大升高。
- 在 V_4 的 B 极(图 5.2 中的信号点 C)，对地增加一个 100 kΩ 的电阻，和 A、B 两点之间的绝缘电阻进行分压。由于 100 kΩ 的电阻远远小于 A、B 两点之间的绝缘电阻，所以，现在 A 点泄漏过来的电流所产生的电压就很难让三极管 A_4 轻易导通。
- 这样处理后，三极管 A_4 的导通只受控于集成电路 W91330A 的第 13 脚。
- 用以上 2 个手段处理后，即使在电路板很潮湿的时候，外面即使加入远高于国标的 120 V 电压的振铃，也不会再有“电话呼入后，响一声就没了”的

故障了。

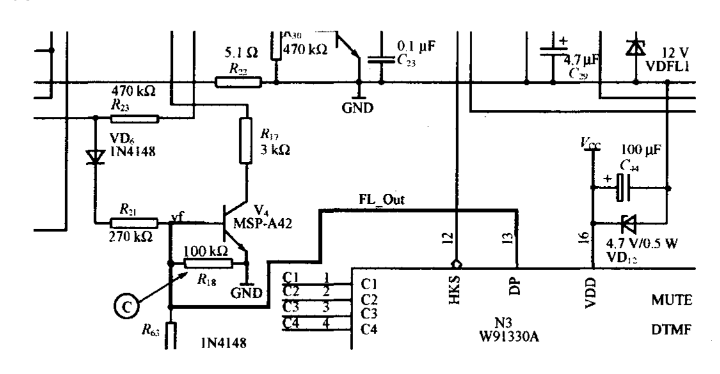
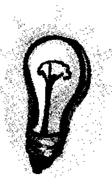


图 5.2



● 诱导法，比较适合查找那些在极端状况下才出现的问题。比如电压太高或者太低、温度太高或者太低、湿度太高或者太低的状况。

● 极端状况下才出现的问题通常都是出现概率比较小的故障。查找故障很费时间，所以制造条件诱发故障，是节约时间的一个重要方法。

5.2 面对黑箱的对策

面对一个模块，而无法得知其内部具体结构的时候，比如一个 IC、一个不开源的软件库、一个看不懂代码的软件模块、一个电路模块等。这些东西经常根本找不到描述它们内部的资料。这个时候，你面对的就是一个黑箱，一个你不知道里面是怎么来、怎么去的黑箱。

但是对付黑箱也是有办法的。就像古代中医，他们并不知道人体内部的那么多细节（比如血压、血液里面的各种成分、内分泌的运作流程等等），不可能拿出对策去降低血压，也不可能开药去精确调节一个人的内分泌系统。但这并不影响中医之伟大。这是为什么呢？这就是因为任何一个黑箱，它总是有输入条件，也总会有输出的现象。通过对输入的改变和调整，可以影响输出的结果和状态。中医虽然对人体内部的信息知之甚少，但是通过“望闻问切”可以知道该患者有哪些输出，比如舌苔、气色、脉像等，也可以知道有哪些输入，比如饮食习惯，生病之前接触了哪些东西等。在大致知道人体的运行规律，而不能精确得知其

内部结构和微观运行规律的时候，可以通过调节病人的输入条件（比如建议病人改变生活习惯、输入特定的汤药等），来改变病人的输出结果。

这个问题如果用数学模型来描述，应该如图 5.3 所示。

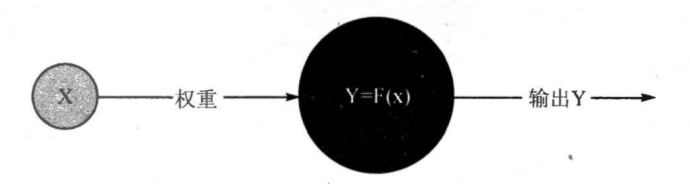


图 5.3

这是一个单输入、单输出的黑箱。这种黑箱比较简单，也比较容易搞定。但不幸的是，经常要解决的问题却是如图 5.4 和图 5.5 那样的多输入条件的黑箱。不仅多输入，而且每个输入还有不同的权重。黑箱把这些不同权重的东西，经过未知的处理流程之后得到一个（图 5.4 ）或者多个（图 5.5）输出。这个处理流程可以理解为一个数学公式，而这个数学公式我们不知道而已。

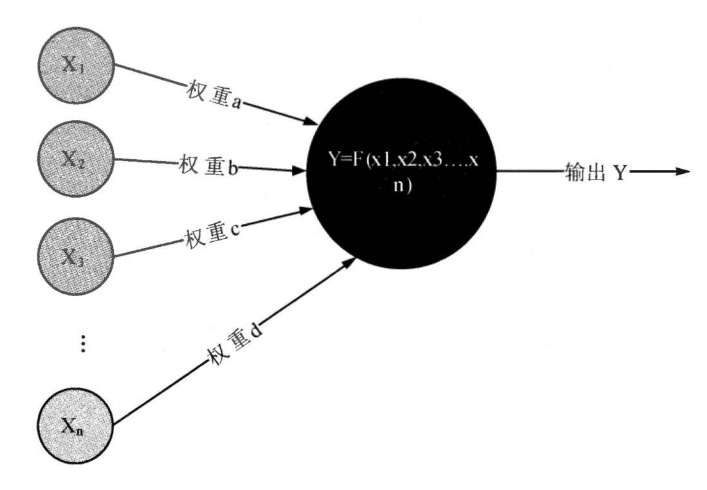


图 5.4

面对这样的黑箱的时候，要做如下的事情：

1. 搞清楚正常的输出结果应该是什么样的。
2. 故障时候的输出是什么样的。
3. 故障发生的时候，各个输入条件是什么样的。
4. 故障时候的各个输入条件的权重是什么样的。
5. 尝试着调节输入条件 X_1、X_2…、X_n 及其对应的权重。

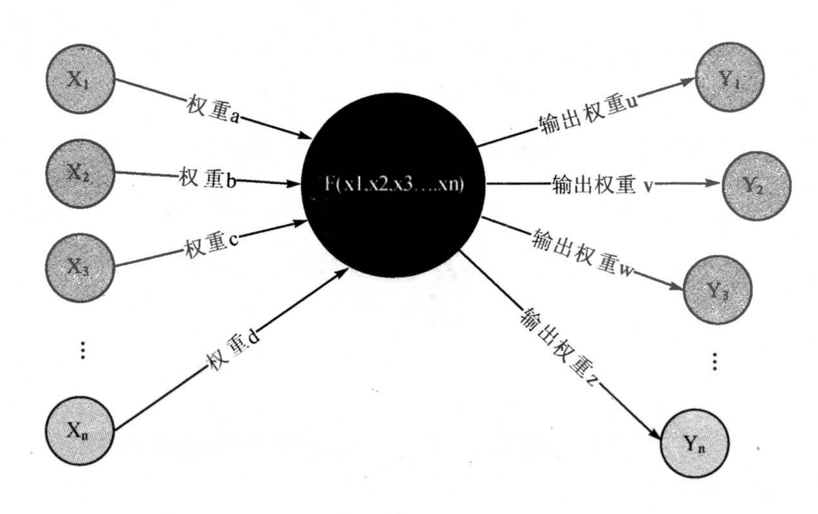


图 5.5

6. 调节之后观察各个输出结果和各个输出结果对应的权重,估计或者计算这些输出结果和你需要的结果有多少差距。
7. 根据这个估计出来的差距,反过头来再次调整输入条件 X_1、X_2…、X_n 及其对应的权重。
8. 重复步骤 5～7,直到各个输出结果和各个输出结果对应的权重达到或者接近你需要的那个值。

以上流程,其实就是控制论所说的“闭环控制”。

抛开电子工程,拿一个中医诊断腹泻的例子来说明这个问题,可能更好理解。大家知道,腹泻有几种可能:

1. 食用了不干净的食品(如果没有,权重就是 0,否则就是权重过大。以下类同)。
2. 食用了干净但是比较油腻的食品。
3. 腹部受凉。
4. 服用了一些有一定毒性的药品,或者一些药品的剂量超过额定剂量(也就是权重过大)。
5. 发病前的人体体质,也会严重影响腹泻的发生概率,所以它也是一个输入条件。如果身体太虚弱,则是权重太低。
6. 我们不知道人体内部精确的运作过程,但是可以大致的知道:腹泻产生的概率和条件 1～4 的权重成正比,和条件 5 的权重成反比。

以上这些输入条件,经过人体这样一个黑箱子处理后,就有可能产生腹泻的

输出结果。而人体产生腹泻的输出结果的同时，还会输出一些其他的信息，比如脉象的改变、舌苔的改变、人的精神状态的改变、血检报告单中某些指标的改变等。而治疗过程，就是找到是由于图 5.6 中的哪个输入条件导致的腹泻，然后综合改变各个输入的权重达到治疗的目的。比如食用不洁食品导致的腹泻，则处理结果就是立即停止食用不洁食品(即食用不洁食品那条路径的权重更改为0)，同时加大药品的服用剂量(即药品那条路径的权重加大)。比如是着凉引起的腹泻，则可以立即去除着凉的产生条件(即腹部受凉的那个路径的权重更改为0)，而别的输入条件的权重，不用加大也可以解决问题。

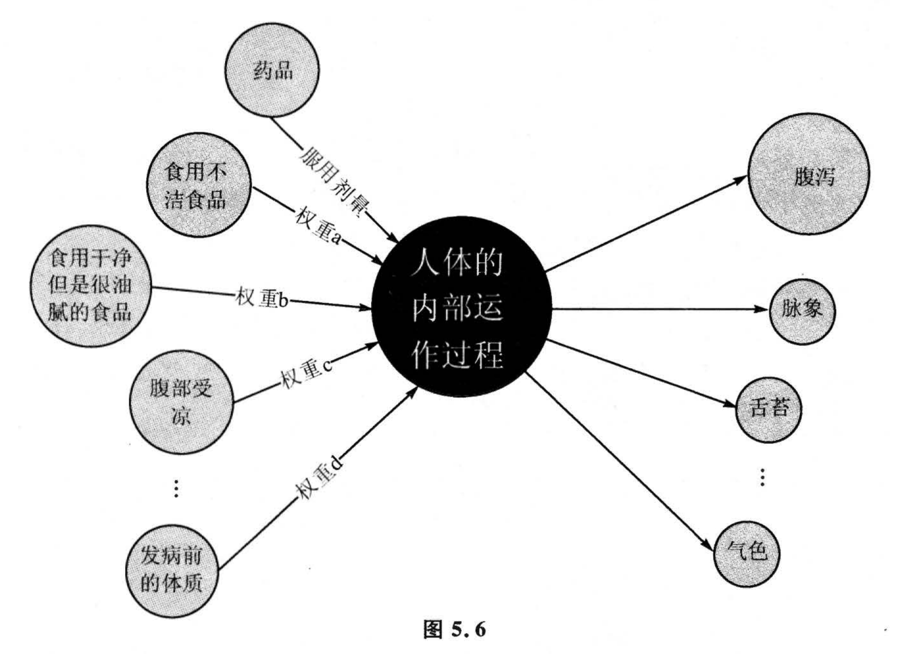


图 5.6

从前设计过 MODEM，使用 TI 的 DSP 芯片。TI 同时提供该 DSP 的 MODEM 解码软件核心。但是在试制的过程中，发现 MODEM 经常掉线。线路质量稍微差一点点，就会掉线。后来联系 TI 公司的 FAE，FAE 提示我们可能影响掉线率的地方(见图 5.7)有：

- 模拟通道的电源纹波。
- 电话线收发回路的消侧音如果不好也会导致接收噪声增大而掉线。
- 实在不行可以更改软件代码中的一些关键参数，但是 FAE 自己也不知道具体是哪个参数，需要自己找。

在掉线之前，误码率一定会上升到一定程度。

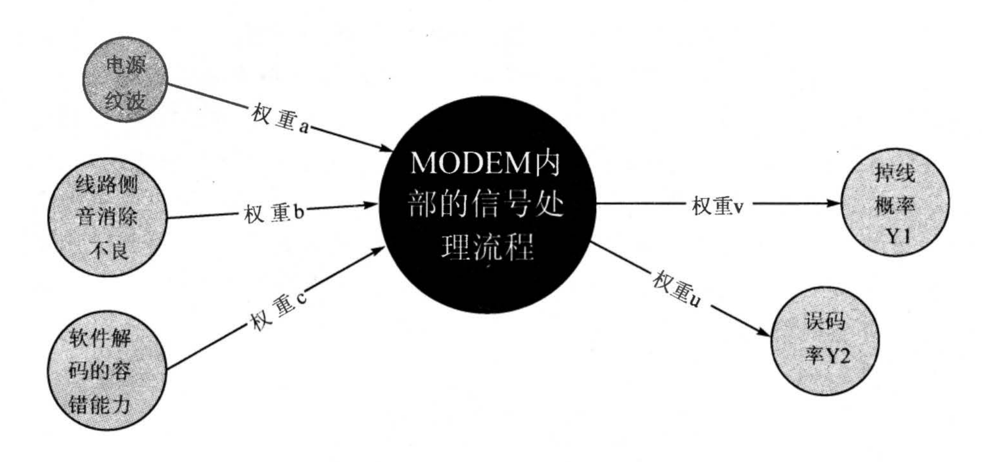


图 5.7

根据其提示，首先试图更改软件代码中的一些关键参数，但是这个软件源代码实在太多了，而且这些源代码是基于一大堆一时半会不能理解的数学原理所写的。要逐个代码去理解，实在费时间。而且这个软件是国际知名公司提供的，就冲着那个品牌，也宁可相信它内部的问题不会太多。于是先把软件部分当作一个黑箱子，先从硬件入手。首先按照本书的文章“搞定电源”中的方法，排除了电源纹波。之后又精确调整电路的消侧音电路。经过这两项调整后（见图 5.8），误码率大幅度降低，掉线的问题也解决了。

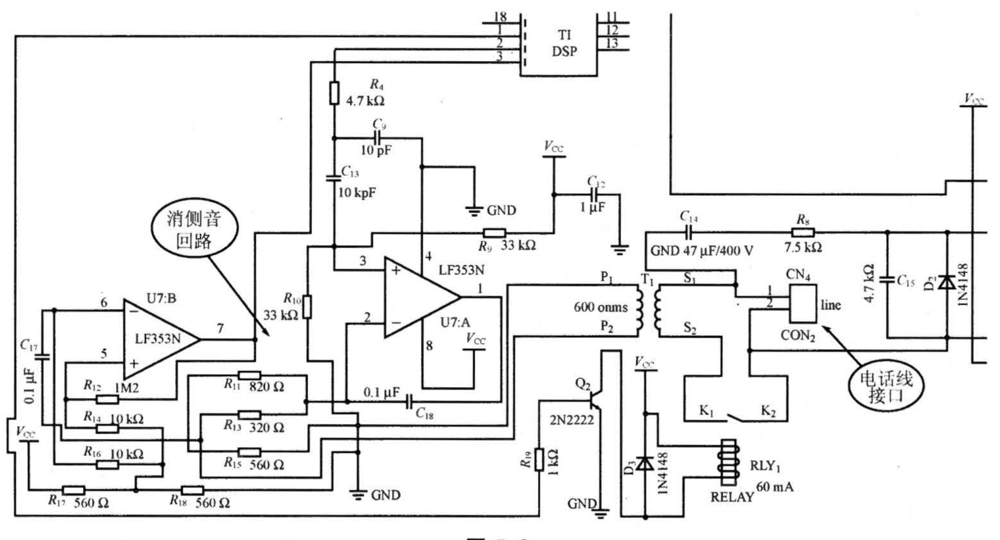
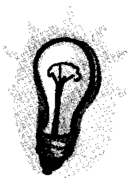


图 5.8









提示：

- 黑箱法也有不适用的时候，比如那些内部已经严重损坏，已经丧失运转能力的黑箱。这种黑箱，无论如何调节其输入参数和权重，都不能让黑箱回到稳定状态。这时，还是要拆开黑箱，用下面的“替换法”和“逐层分割法”来追查故障点最合适。
- 黑箱法最适合于那些内部结构合理，内部没有重大毛病的系统。它虽然有毛病，但是基本还能运转，而且这种故障只是由于输入条件的不合理造成的。这个时候用上述的黑箱法最好。
- 黑箱法只要控制好几个关键参数就可以搞定问题。这样比深入黑箱省时省力得多，这是它的最大好处。

5.3 几种常见的处理故障的方法

5.3.1 替换法

替换法最好理解：怀疑谁坏了，就换掉谁。这是最容易理解和可操作性最好的方法。替换法是用好的部件去代替可能有故障的部件，以判断故障现象是否消失的一种维修方法。好的部件可以是同型号的，也可能是不同型号的。但只要接口相同、功能类似就可以在查找故障的时候临时或者永久性地替换。

替换的原则通常是 2 个：

- 从部件的故障率高低来考虑最先替换的部件。故障率高的部件先进行替换。至于元器件的故障率的高低，可以参考本书的“电子元件故障发生概率排行榜”一章。
- 按先简单后复杂的顺序进行替换。替换工作简单，不怎么费时间的，要优先进行。

比如，有些人家里电视机在接收有线电视的时候有条纹，这个时候就可以用 DVD 机的视频（video）信号输入到电视机中，如果没有条纹，就说明电视机的基带视频部分是好的，但是还不能完全排除电视机有故障的可能，因为视频（video）信号输入正常并不能排除电视机的高频头、中放等部分故障的可能性，这个时候，如果有一个射频（RF）的信号输入就可以解决，这样的信号，我们家里经常也

有，比如一些游戏机就有射频(RF)输出端，用这个输出端连接到电视机的射频(RF)输入端，就可以完全确定电视机的好坏。这个方法其实就是替换法的应用。

5.3.2 逐层分割法

图 5.9 是我曾经做过的一个设计，这个电路图中左边是一个 LM358 运算放大器，做预放大用，右边是一个 LM386 功率放大器。一次有几个样品的喇叭中(也就是图中的信号点 A)有"嘶嘶嘶"的杂音，虽然很微弱但是也很讨厌。为了解决这个问题，首先需要找出干扰源在什么地方。我是这样做的：

- 在信号点 A 连接一个示波器，示波器置于交流挡，时间轴置于每格 50 μs 档位。可以看见"嘶嘶嘶"杂音所对应的电信号，这是一个电压峰值为 100 mVp－p 的一个宽频噪声信号。
- 首先怀疑是不是电源端窜入的纹波导致的问题，所以在 C_2 两端分别并联一个 1 000 μF 电解电容和一个 104 电容，以降低纹波。但都解决不了问题。
- C 点是来自 CPU 的键音信号，这个数字信号从 CPU 出来的时候是 5 V 的数字信号，经过 C_{46}、R_{36}、R_{40} 衰减后输入到 LM386 的"－"输入端。为了确认是不是这一路引起的噪声，我把 C 点的输入信号去除(拆除 C_{46})。等于是完全去掉了这个输入端的信号，但是噪声依旧。
- 用同样的方法，拆除 R_{37}，将信号点 D 的信号去掉，噪声立即消失。说明故障点在信号点 D 的左边。

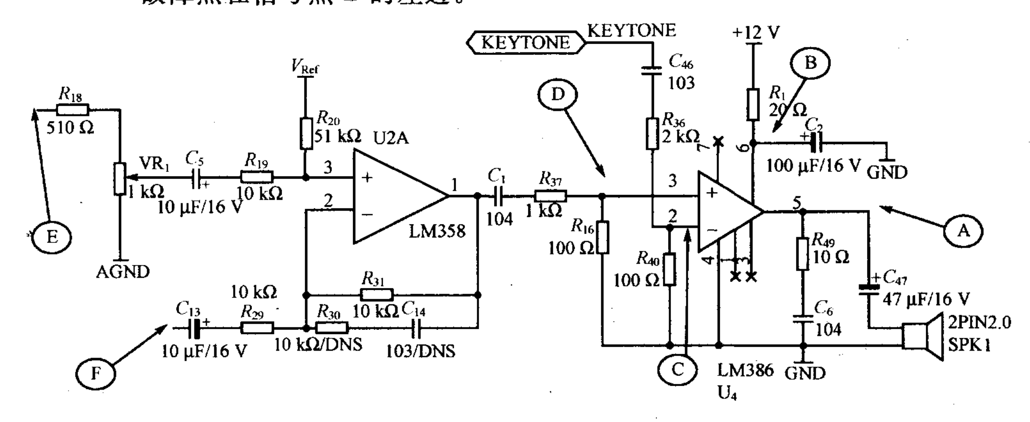


图 5.9





- 在信号点 D 前面有 2 个输入信号 E 和 F，分别接到运算放大器 LM358 的正负输入端。
- 再将信号点 F 的信号必须经过的 C_{13} 拆除，噪声也立即消失。
- 继续在信号点 F 之前查找噪声源，很快就能排除这个故障。

提示：

- 以上的思路就是层层分割，逐步缩小包围圈的办法。每次去掉一条可能的噪声干扰路径，所以很快就能精确定位到故障点。
- 这个方法是查找故障最经常使用的方法。

5.3.3 搜索、穷举法

前几个月，我去做第二代身份证，在拍摄照片的时候，看见那个辅助拍照软件，在人像的位置上，先画好了一些线段。拍照的时候，人像的五官需要在规定的位置，照片才算合格，所以经常为一个人拍照需要耗费 10～15 分钟。很多人搞不明白为什么拍照要这么复杂，我想可能是这样：通过这些辅助线拍摄的照片，便于计算机进行处理。可以用计算机为每一个公民建立人脸的特征数据库，有利于公安系统的管理。你可以想象一下，如果某人抢劫银行，被摄像头拍摄下来了，那就可以根据这个拍下来的人脸，在相关的数据库中搜索，是不是一下就可以穷举出很多可能的嫌犯？然后把这些嫌犯逐一排查，是不是就可以大大加快破案的速度？这就是穷举法的另一个用途。

一次维护一个离职同事写的代码，其中电路板上有一段配置程序，它在串口上的输出是这样的（以下为了简要说明，特删去无关的显示内容和代码段）：

TinyTCP/IP Config Application (v0.0.23 beat, May 20 2006)

1: Change serial number: 0

此处省略……

8: Disable DHCP & IP Gleaning: DHCP is currently enabled

9: Download fileimage.

0: Save & Quit.

Enter a menu choice:

本来选择 0～9 都应该有对应的反应，但是这个软件输入 9，立刻又把这个菜单再次显示一遍，一点都不像执行了命令，想了很多种原因都不能解决，估计问题就在软件里面了。由于该同事离职太久，联系不上，只好自己找原因。拿出

他的源代码，这个源代码有 7 000 行之多，要是逐个模块去找，要费很多时间。所以用编辑器一下打开这个软件工程的所有源代码(21 个文件)，在这 21 个文件里面搜索关键字“Download fileimage”，结果就快速定位到了以下的一段代码：

```
static void SetConfig(void)
{
            此处省略……
    while(! bQuit)
    {
        // Display the menu
        putrsUART("\r\n\r\n\rTinyTCP/IP Config Application (v0.0.23 beat, May
        20 2006)\r\n\r\n");
        putrsUART("\t1: Change serial number:\t\t");
        wvTemp.v[1] = AppConfig.MyMACAddr.v[4];
        wvTemp.v[0] = AppConfig.MyMACAddr.v[5];
        uitoa(wvTemp.Val, response);
        putsUART(response);
            此处省略……
        putrsUART("\r\n\t8: ");
        putrsUART((ROM BYTE * ) (AppConfig.Flags.bIsDHCPEnabled ? "Dis" : "
        En"));
        putrsUART("able DHCP & IP Gleaning:\t\tDHCP is currently ");
        putrsUART((ROM BYTE * )(AppConfig.Flags.bIsDHCPEnabled ? "enabled" : "
        disabled"));
        putrsUART("\r\n\t9: Download fileimage.");
        putrsUART("\r\n\t0: Save & Quit.");
        putrsUART("\r\nEnter a menu choice: ");


        // Wait for the user to press a key
        while(! DataRdyUART());
        putrsUART((ROM char * )"\r\n");
        switch(ReadUART())   // Execute the user selection
        {
            case '1':
                此处省略……
                break;
```

```
            case '8':
                此处省略……
                break;
            此处省略……
            case '9':
                #if defined(FILE_USE_FLASH) && defined(STACK_USE_FILE)
                    Download_file_image();
                #endif
                break;
            case '0':
                此处省略……
                break;
        }
    }
}
```

根据这段代码，可以知道，可能是在箭头所指的那个语句行的 2 个宏定义没有打开。再次以“FILE_USE_FLASH”和“STACK_USE_FILE”作为关键字在这 21 个文件里面搜索，一下就找到了这 2 个宏定义放置的地方，原来是被作者注释掉了，去掉注释后，这个第 9 选择项对应的功能就 OK 了。

再举一个例子，图 5.10 是某个单片机内部的某个模块的框图，右边的 CCP1/P1A、P1B、P1C 和 P1D 是单片机的 4 个 pin，我有一个故障和 CCP1/P1A

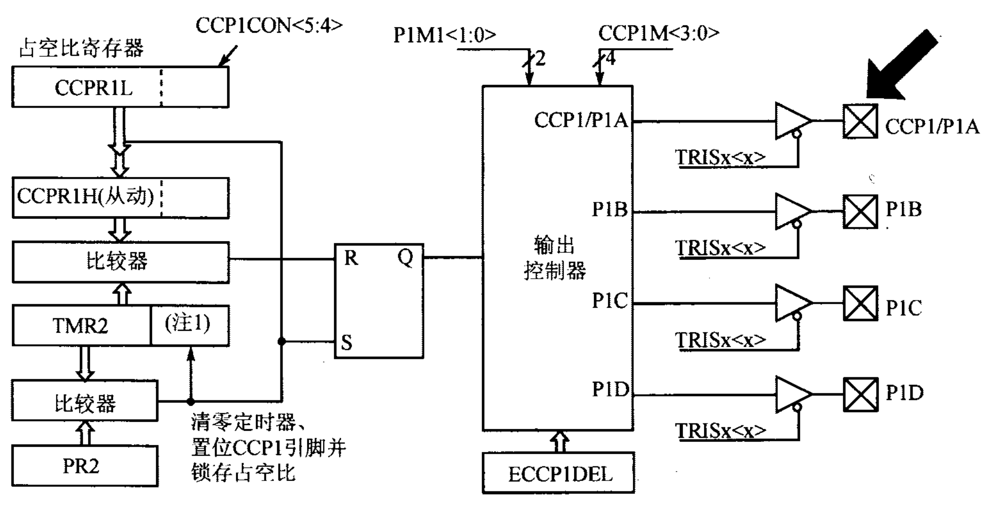


图 5.10

引脚有关，这个故障的发生可能是软件牵扯到了别的什么东西，因为单片机的引脚经常是复用的。为了弄清楚单片机什么地方和这个CCP1/P1A引脚有关联，当然可以在整个IC的规格书里面去找，但是这个规格书有700页之多，别说一页一页仔细地去找，就是在电脑上用PageUp/PageDown一页一页把规格书翻一遍也够呛。所以我这样做：

首先选择Acrobat软件的搜索功能，如图5.11所示。

然后弹出这样的窗口（见图5.12），在其中填入字符串“P1A”，为了屏蔽一些无关的搜索结果，我选择了“全字匹配”和“区分大小写”。

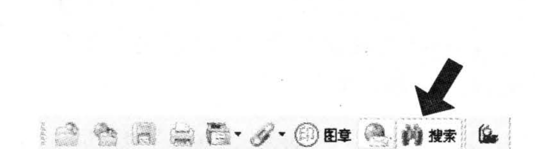


图 5.11

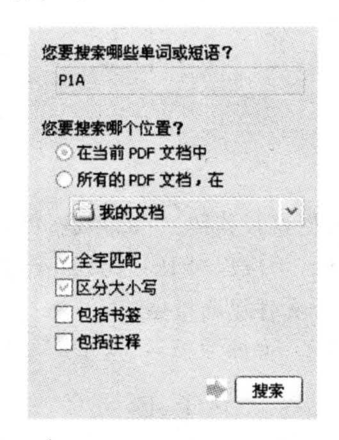


图 5.12

开始搜索后，出来图5.13的搜索结果。按照这54个搜索结果，逐一排查，很快就能定位到故障的发生地点，免去了在整个PDF文档里面查找的麻烦。



- 充分使用资料也是搞定故障的一个重要方法。
- 搜索法在拥有完整资料，且资料是电子版的时候最好用。
- 搜索法不仅可以在一堆资料中快速定位到相关的故障关联点，也可以在一堆资料中快速穷举出你所关心的信息点。
- 搜索和穷举法如果用人工来完成，是一个很笨的办法，但是用计算机来完成，却是很轻松很神速的事情。

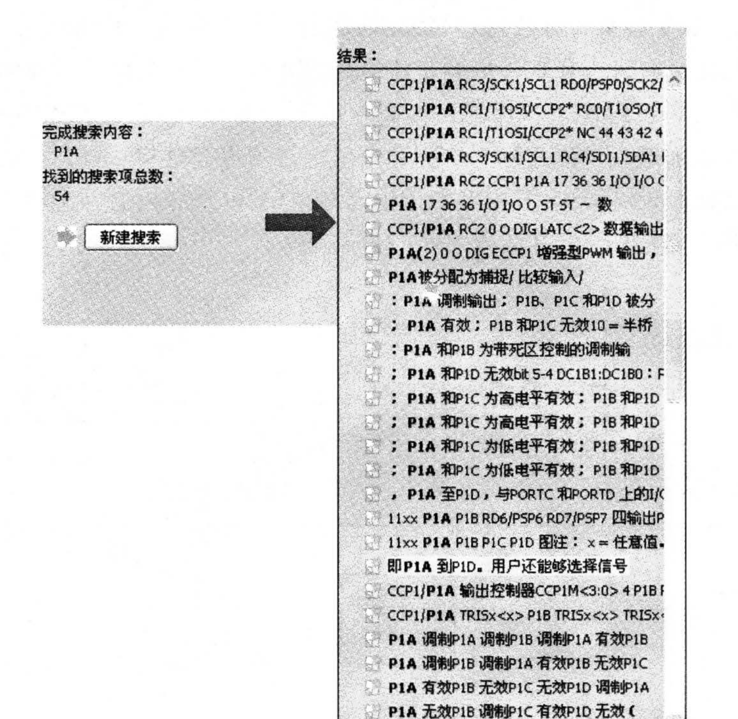


图 5.13

5.3.4 计算法

以前设计过一个光端机，就是那种一个光纤进去，N 个数据链路出来的那种机器。设计的时候，有几个样机老是会莫名其妙地告警，有时候半小时出现一次，有时候半天一次，最长的时候连续运行 3 天才出现一次。而满足告警的条件是：

1. 数据链路 $L_0 \sim L_3$ 的信号消失。
2. 数据链路 $L_0 \sim L_3$ 的误码率达到一个规定的上限。
3. 数据链路 $L_0 \sim L_3$ 的码流中包含告警的指令。

这个机器与此故障的相关部分简化为图 5.14，CPU 通过 SPI 访问数据链路管理芯片，得知是否有告警故障发生。图 5.14 的电路模块 A 和电路模块 B 是完成别的功能的。反复排查了以上三种可能性，都不能成立。看来是自己的设计问题。

我们尝试了很多种办法，试图发现问题所在。后来用万用表无意中发现电路中的 A、B 两点居然有 0.3 V 的直流电压，用示波器观察，除了直流电压外，由于 CPU 和数据链路管理芯片都在很高频率运行的缘故，还叠加了很强的交流噪声，二者叠加最大可以达到 0.5 V 的峰值。之所以会这样，是因为电路板上的地线很细，而且没有覆铜，且电路板上电路模块 A 和 B 的耗电也比较大，这样进一步推高了电路板上 A、B 两点之间的直流压降。

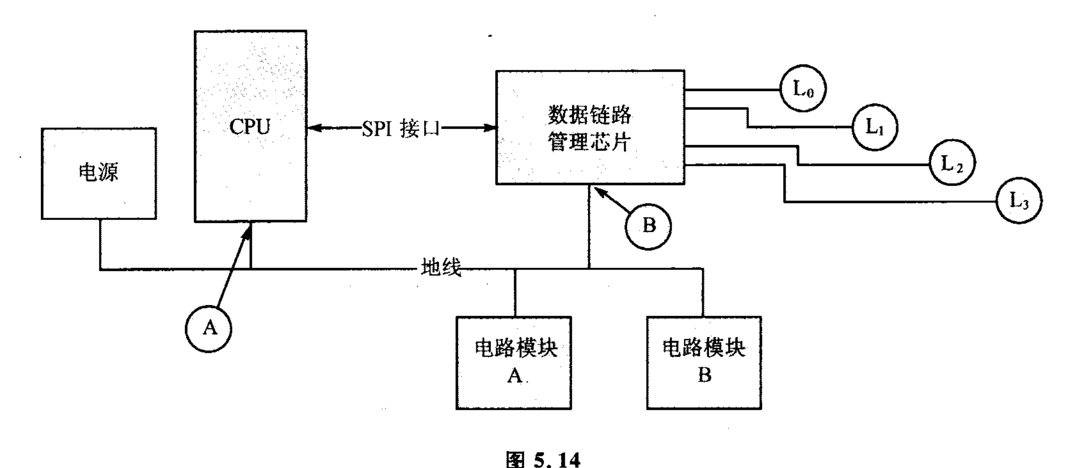


图 5.14

可以看看图 5.15 中这个 CPU 的输入条件，图中 B 指示的是它最大允许的 0 电平输入电压，就是 $0.2*V_{CC}-0.1$，即 $0.2*5\ \text{V}-0.1=0.9\ \text{V}$。而在图 5.16 中则指出了这个数据链路管理芯片的输出 0 电平的情况，这里它的输出电压最高可能有 0.35 V，那这样加上地线上 0.5 V 的峰值，就有 0.85V 的电压差。距离图 5.15 中 CPU 的 B 点电压就差一点点。所以，如果再出现什么异常信号（具体什么异常信号，我不知道。因为它稍纵即逝。电路中的干扰信号，总是无时不在、无处不在的，不同的只是出现概率、电压大小和频率的差别。）这个 0.85 V 的电压就会被进一步推高，达到 CPU V_{IL}参数的上限。所以 CPU 从 SPI 端口读数据链路管理芯片内部的告警寄存器就有出错的可能。这个地方明显是一个不安定因素，因为这个地方刚好被设计在了临界状态，所以平常工作都是好的，很久才会出现一次问题。

解决的方法很简单：

- 那几个出故障的机器，已经安装好了，扔掉可惜。我们就把一根粗一点的铜线，一路顺着电路板上的 GND 走线，焊接过去。每隔 10～15 mm 把铜丝焊接到电路板上的 GND 铜箔上，以降低电路板 GND 走线的整

DC ELECTRICAL CHARACTERISTICS

T_{amb}=0 ℃ to+70 ℃; V_{CC}=5 V±10% or–40 ℃ to+85 ℃;5 V±5%; V_{SS}=0 V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN (A)	TYP[1]	MAX (B)	
V_{IL}	Input low voltage	4.5 V< V_{CC} <5.5 V	–0.5		$0.2V_{CC}-0.1$	V
V_{IH}	Input high voltage(ports 0,1,2,3, $\overline{EA}$)		$0.2V_{CC}+0.9$		$V_{CC}+0.5$	V
V_{IH1}	Input high voltage,XTAL1,RST		$0.7V_{CC}$		$V_{CC}+0.5$	V
V_{OL}	Output low voltage,ports1,2,3[8]	V_{CC}=4.5 V, I_{OL}=1.6 mA[2]			0.4	V
V_{OL1}	Output low voltage,ports 0,ALE, $\overline{PSEN}$[7,8]	V_{CC}=4.5 V, I_{OL}=3.2 mA[2]			0.45	V
V_{OH}	Output high voltage,ports 1,2,3[3]	V_{CC}=4.5 V, I_{OH}= –30 μA	$V_{CC}-0.7$			V
V_{OH1}	Output high voltage(ports 0 in external bus mode), ALE[9], $\overline{PSEN}$[3]	V_{CC}=4.5 V, I_{OH}= –3.2 mA	$V_{CC}-0.7$			V

图 5.15

体电阻，来减小压降。这样处理后，机器就固若金汤了。

● 由于这是开发阶段的电路板，下一个版本的电路板注意地线的走线和各个芯片的退耦就可以了。

Reset Voltage	V_{RST}	–	0.4	0.7	1.0	V
Reset Current	I_{RST}	–		0.1	0.4	mA
Low Output Voltage	V_{OL}	V_{CC}=9 V I_{SINK}=10 mA I_{SINK}=50 mA	–	0.06 0.3	0.25 0.75	V V
		V_{CC}=5 V I_{SINK}=5 mA	–	0.05	0.35 (C)	V

图 5.16

还有一个案例，请看图 5.17，这是一个麦克风放大的电路。麦克风放大后的输出信号总是失真。示波器观察这个输出信号，发现输出的语音信号底部被切割掉了，所以造成声音很难听。但是用万用表去测量运算放大器的输出端，它的输出直流电位只有 3 V。而按照运算放大器的常规，输出电压应该和参考电压等同(图 5.17 中的参考电压 $V_{ref}=6$ V)。按照逐层分割法的思路，检查如下方面：

● IC 供电电压，12 V，正常。

● IC一输入端，电压和输出端一样，也只有 3 V。

● IC一输入端的所有元件，都无异常。

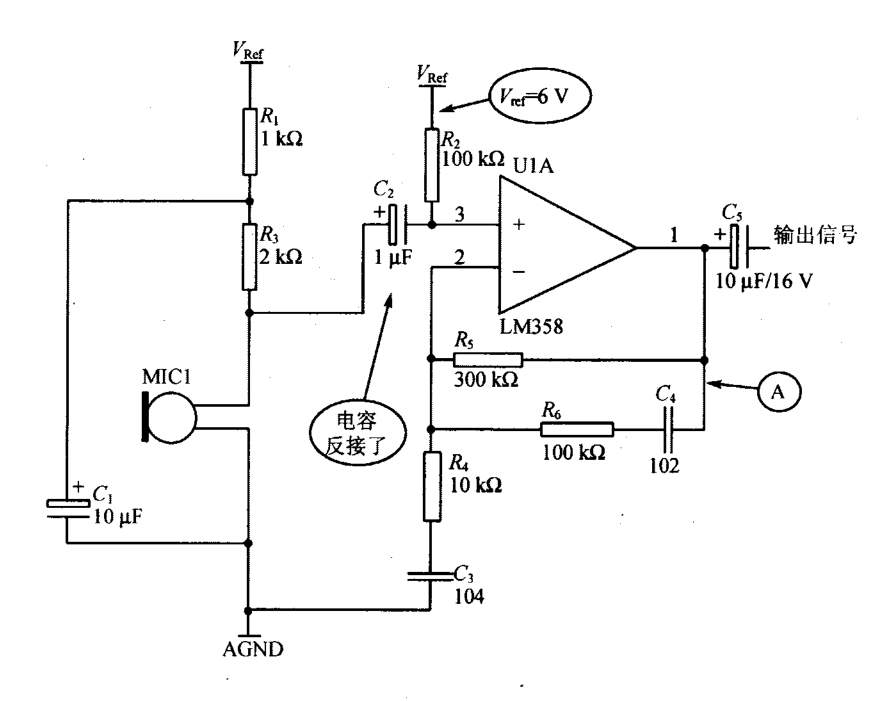


图 5.17

- IC+输入端,电压和输出端一样,也是 3 V,这意味着 R_2 两端产生了 3 V 的电压降,通过其间的电流为 3 V/100 kΩ=30 μA。
- 而运算放大器的输入阻抗都很高,不可能会有 30 μA 的电流进入 IC。
- 所以这 30 μA 的电流就有可能是通过了电解电容 C_2。
- 仔细一看 C_2,是被焊接反了。而麦克风的上端的电压是 2 V。
- 电解电容反接后,其漏电阻很大。把 R_2 的偏置电压分压了一部分,造成 IC+输入端电压下跌。
- IC+输入端电压下跌,所以输出端的电压也跌到了 3 V。
- 信号大一些的时候(振幅超过±1.5 V),运算放大器内部的下拉输出三极管就饱和了,所以就在如图 5.18 所示的示波器上看见语音信号底部被切割掉了。

再说一个案例:一个熟人打电话问我,说有一个语音设备,其按键背景灯由 12 V DC/DC 供电,但是随着音乐的起起伏伏,按键背景灯老是会随着音乐亮度明显变化,并且传真给了我一个电路图。我把它整理出来,去掉无关的,如图 5.19所示。我看完这个图,打电话问他一个问题:“随着音乐的起起伏伏,DC12

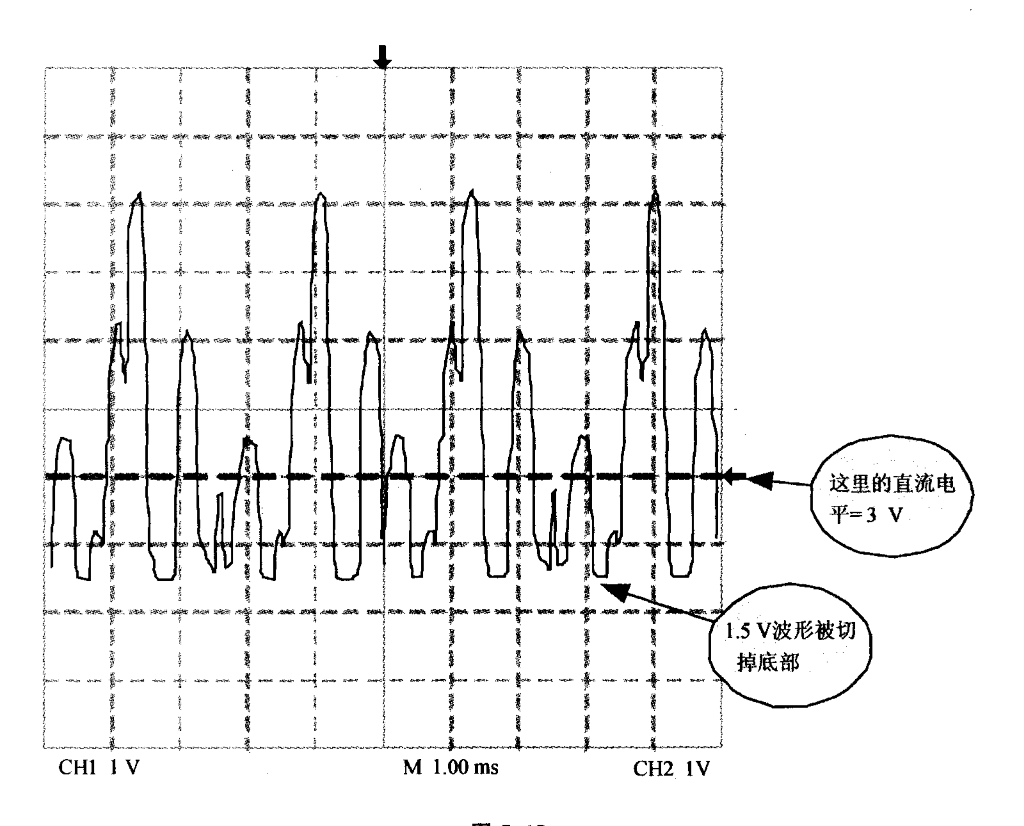


图 5.18

V 会变化吗？变化多少?”。他测量了一阵子后回答：“会有电压变化，不过很少，才 0.3～0.4 V 左右。音量大的时候，电源会掉下来!”。我建议他先去看看电源是不是有什么问题。2 天后，告诉我：“这个 0.4 V 的跌落，死活搞不定!”

后来我仔细想了一下，他死活搞不定这个 0.4 V 的电压跌落，其实也很合理。因为他的这个 12 V DC/DC 供电本身就不是大容量电源。当音频功率放大器在大音量输出的时候，音频功率放大器消耗的电流势必大增，电压往下跌落一点，也很合理。只是他的这个 12 V DC/DC 的跌落太多了一点(一般 12 V DC/DC 在大电流输出的时候，电压跌落 0.2 V 是十分正常的)。然后我计算了一下，马上就发现了问题所在：

- 根据我的经验，一般红色 LED 发光的时候，其正向压降都在 1.8 V 左右。
- 6 个 LED 的压降是 6 * 1.8=10.8 V。

- 那限流电阻上的压降是 12－10.8 V＝1.2 V。
- 限流电阻上的电流是 1.2/510＝2.352 mA。
- 如果电压跌落，根据 LED 的特性，只要它还亮着，那它的正向压降基本就不会改变。所以电压跌去 0.4 V 的时候，限流电阻上的电流是(1.2－0.4)/510＝1.568 mA。
- 1.568 mA/2.352 mA＝66%，一下电流少了 33%。
- 所以，按键背景灯老是会随着音乐的起伏而亮度明显变化。

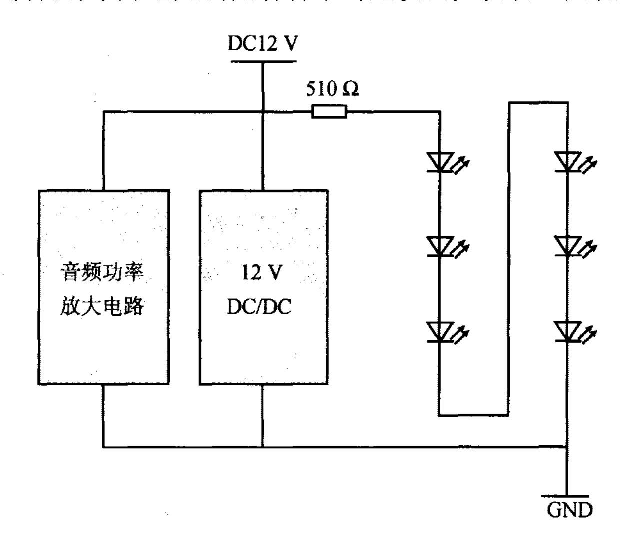


图 5.19

然后我给他的建议是，不要去修改电源，因为 12 V DC/DC 电源跌落 0.4 V 似乎也在合理的范围内。我建议他更改成图 5.20 的方式。这样改动之后每个 LED 的电流就是(12 V－1.8 * 3)/1 kΩ＝6.6 mA。当电压跌落 0.4 V 的时候，是(12 V－0.4 V－1.8 * 3)/1 kΩ＝6.2 mA。电流只会减少(6.6 mA－6.2 mA)/6.6 mA＝6%，变化率只有原先的 1/5 不到。他按照我的处方更改电路后，基本上就看不出来键盘背景灯还有原先的那个问题了。

我处理这个故障的时候，手上始终没有拿到他的电路板，只根据电路图和电路理论计算他的电路的参数，不仅很快找到问题症结所在，而且很快就解决了问题。



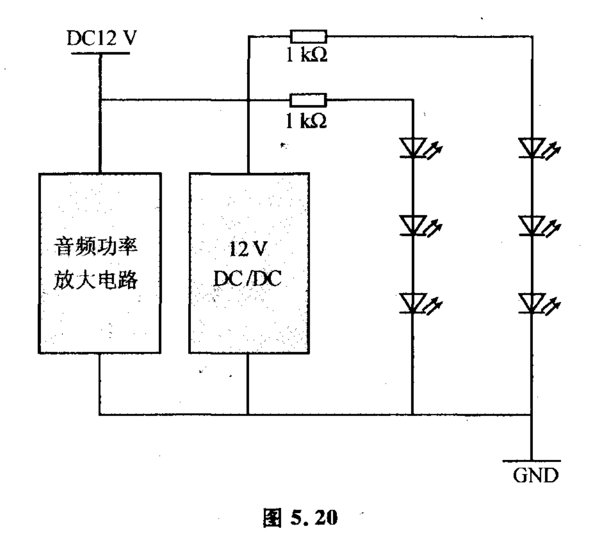


图 5.20

5.3.5 数据统计法

古时候，经常发生很多烈性传染病；由于科学知识不发达，人们对疾病束手无策。所以经常一次疫病的爆发，一个城市的大部分乃至所有的人口都会死亡。比如天花，从有记载的历史开始，到完全控制，累计死亡人数超过 10 亿之多。而人们明白天花的病原体结构，也不过是最近 100 年内的事情。人们成功抵御天花的方法，却在我们的宋朝就有文字记录了，之所以没有广泛推广，只是因为当时的那些方法还不够安全。那时的人们，虽然并不懂其中的道理，但是通过几百年的长期观察和记录，人们逐渐发现和确认了：得过一次天花的人，就不会得第二次天花。也就是基于这样的观察，所以才有了后来多种形式的“种痘”。直至 20 世纪 70 年代末，在地球上彻底控制了天花。1980 年世界卫生组织（WHO）宣布了这一结果，并在全世界停止了普遍种痘。停止种痘 20 多年来，世界上没有发现一例天花病人。发现天花的运行规律，在当时的科技水平下绝对是伟大的工程。而我们的设计，通常没有这么复杂。所以我想通过这个例子告诉大家的是：即使你不明白设计为什么出问题了，也可以通过细致的、不懈的观察来发现问题的内在规律，从而拿出解决问题的办法。

20 世纪 50～60 年代，医院经常出现一些耳聋的儿童，比平常的年份，出现的百分比高了很多。开始几年，谁也不知道这是为什么。后来一些卫生组织对这些耳聋儿童的病历进行汇总，总共汇总了 2 000 多份病历，进行详细的数据统计和分析。筛除掉一些传统的可以导致耳聋的因素后，结果发现这些患儿很多

使用过链霉素，而且剂量都偏大。虽然就整体而言，链霉素导致耳聋的概率不算很高（这也就是为什么早期临床试验发现不了这个副作用的原因），但是一旦大面积使用，就会有一部分人出问题。通过对大批数据的统计和分析，人们在不知道其导致耳聋的机理的时候，就发现了链霉素和耳聋之间的关联性。

原先设计的电话机，有一个型号量产了。开始一直都很正常，后来有一阵子，经常会有免提啸叫的问题。出现的概率大约是 7%～9%。而我们正常的情况大约为 0.1%～0.2%。为了搞清楚这个问题。把故障机器拿一批次来进行检查，单单看电路板，丝毫看不出什么名堂。为了进一步搞清楚问题出现的原因，把这 30 台故障机的电声性能进行了一个完整的测试，输入标准的测试信号后，记录下每一个故障机的输出数据。为了节约篇幅，这里只写它们的平均值。然后和我们原先试制时候的标本机器进行对比，得到以下表格：

在表 5.1 所列的数据中，可以比较明显的看出，样本机的高频频率响应比较弱，而故障机的高频输出电平都偏高一些。也就是根据这张表格，才知道为什么它们的啸叫声音也比以前的故障机的啸叫声音频率要高。以前的机器啸叫多是低频的"呜呜呜"声，而这批机器啸叫多是高频的尖叫声。有了这个依据之后，着重检查电路的频率响应。

表 5.1

频率/Hz	标本机器/mV	啸叫故障机器的平均值/mV (30 台机器的平均值)
300	800	810
800	1 350	1 370
1 500	1 160	1 250
3 000	710	1 180

这个型号机器的免提部分使用的 IC 是 MC34118，IC 的电路图如图 5.21 所示。依据测试数据和"逐层分割法"的原则，将故障定位在了 IC 的第 10pin 之前。这里只剩下一个麦克风放大器、一个麦克风、一个耦合电路和一个反馈回路。但是并不能在外观上看出来，是哪个元件存在故障。只好把这里相关的 5～6个元件逐个拆下来测量，取得数据后和设计的底稿对比。通过对比发现是图 5.21 中的那个反馈电容的容量在故障机中普遍偏大，造成放大后的麦克风信号中，中频和高频的含量超标。这样电路板套上机壳后，就容易产生声反馈而啸叫。而电路的高频响应比从前好了，所以机器就更容易啸叫在高频率上。

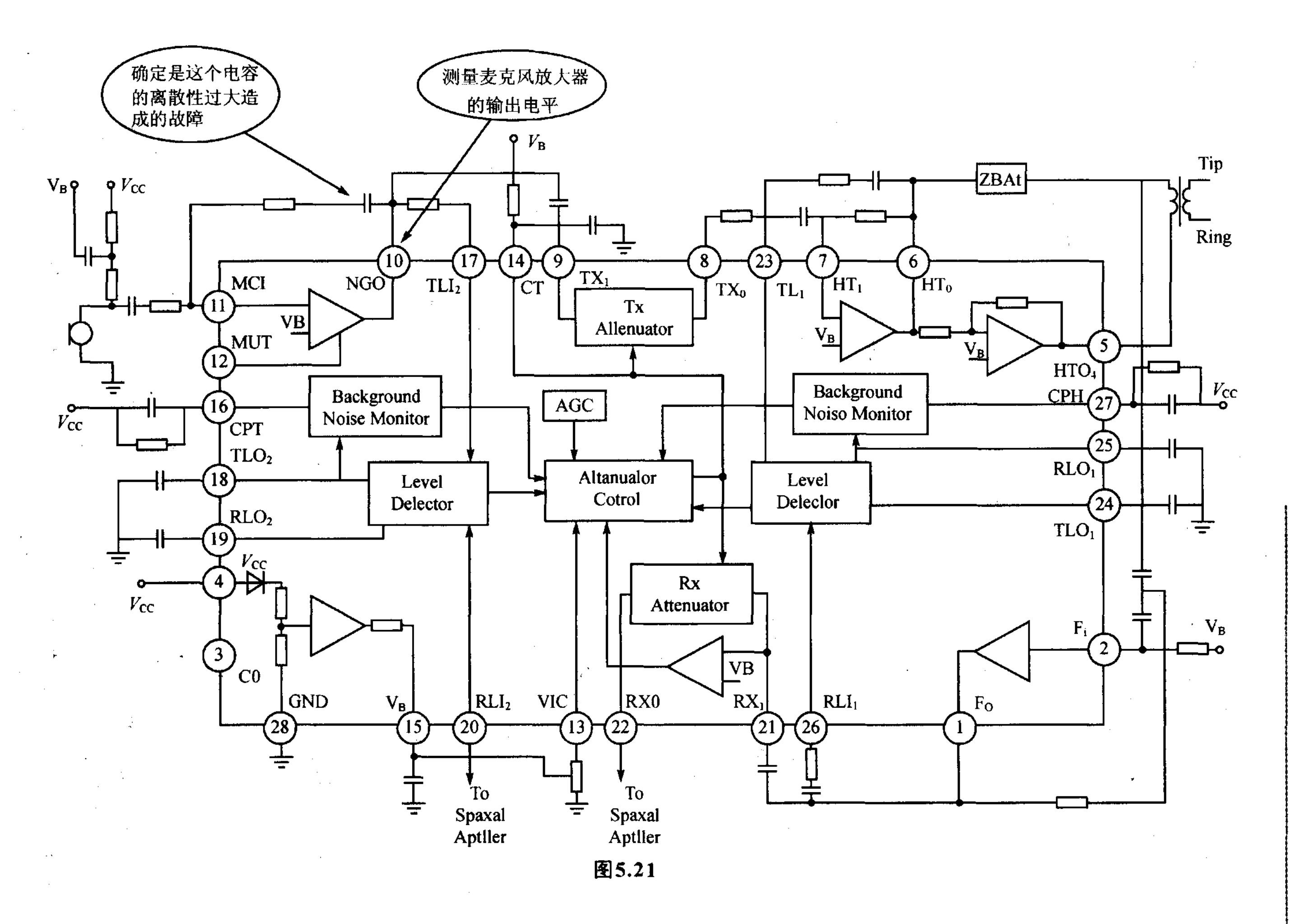

确定是这个电容的离散性过大造成的故障
测量麦克风放大器的输出电平
VB
Vcc
MCI
NGO
TLI2
CT
TX1
TX0
TL1
HT1
HT0
ZBAt
Tip
Ring
MUT
Tx Allenuator
HTO4
CPH
CPT
Background Noise Monitor
AGC
Background Noiso Monitor
TLO2
Level Delector
Altanualor Cotrol
Level Deleclor
RLO1
RLO2
TLO1
Rx Attenuator
Fi
C0
GND
RLI2
VIC
RX0
RX1
RLI1
Fo
To Spaxal Aptller
To Spaxal Aptller

图5.21

这些故障的瓷片电容，虽然标称都是102(10×10^2 pF)，但是把它拿下来后，用仪表测量，普遍都超过了2 000 pF，有的甚至是3 000 pF。对于这样的问题，我们提出的修改意见是：

- 更换这个电容的供应商。
- 对于已经使用这个有故障隐患的电容的电路板，全部拆下来，换成别的厂家的产品，以绝后患。

再举一个例子，我用数据统计法解决数码相机电池的问题。看看我如何在众多电池中剔除老化的电池。

我的一个数码相机，使用4个5号电池。为了使用方便，前后配了3组5号电池。平平静静地用了近一年。之后，随着照相机使用时间的延长，有一种不愉快的现象始终缠绕着我。这个现象就是照相机在使用的过程中会突然关机，我重新开机后又一切回复正常。刚开始，这个现象出现的概率并不大，后来随着时间的推移，这种故障越来越频繁。随着该故障出现次数的增多，我发现有这样的规律：

- 在照相机浏览相片的时候，较少发生自动关机。
- 拍摄模式的时候，发生自动关机的概率大一些。
- 拍摄的时候，如果打开了闪光灯，或者打开了机器后背的外置取景器，则故障发生的概率更大。

依据以上现象，我有些怀疑照相机是否有故障了。但由于故障是随机出现的，无法100%确定故障就是照相机引起的。再进一步的使用，还发现有这样的规律：

- 如果电池刚充好，就在照相机里使用，则出现故障的概率极小。
- 如果电池拍摄了20～30张后，故障开始出现。继续拍摄，故障越来越频繁出现。
- 如果拍摄的时候开了闪光灯，或者打开了机器后背的外置取景器，则故障会提前出现。

依据我平时的经验，再综合以上现象，怀疑有2种可能。一是照相机内部的电源部分有问题；二是电池老化所致。

按照我平常设计时找问题的思路，先做一个实验确认故障是不是电池引起的，数码相机买来后配了2组镍氢电池，一组是4个1 500 mAh品新电池；另外一组是4个1 300 mAh GP电池；还有一组500 mAh的镍镉电池。依据电池的特性，判定一个电池是不是老化的基本方法就是测量它是否经得起大电流放电。我先把3组12个电池分别编上号码，分别是A1、A2、A3、A4、B1、B2、B3、B4、

C1、C2、C3、C4。等3组电池在数码相机上都放电完毕后用万用表测量其开路电压和短路电流，分别得到表5.2所列数据。

表5.2

电池分组	电池编号	开路电压/V	短路电流/A
1500 mAh品新电池	A1	1.330	1.95
	A2	1.325	1.54
	A3	1.325	3.01
	A4	1.332	1.28
1300 mAh的GP电池	B1	1.307	2.05
	B2	1.325	3.20
	B3	1.338	2.99
	B4	1.313	2.11
500 mAh的镍镉电池	C1	1.265	0.10
	C2	1.258	2.54
	C3	1.270	3.04
	C4	1.271	2.55

短路电流的测试方法是，将万用表打在直流20安培的档位上，用万用表的红、黑表笔直接短路电池的正负级，这样做虽然违背了电池使用的安全警告(电池上面一般都写了，禁止电池短路)但只要时间短暂，基本上是没有问题的。出于安全的考虑，笔者不推荐各位读者也用同样的方法去短路1号电池，或者是更大的电池，因为那些电池的短路电流非常大，有可能不到1秒钟就能烧断一根电线。但由于5号电池的短路电流都在5A以下，只要测试5号电池的时候每次短路的时间控制在5～6秒之内，无论对于电池还是对于万用表都不至于造成伤害。

得到以上数据后，再把3组电池用充电器都充满电，重新再做一次这样的实验，得到表5.3所列的一组新数据。

一般电池老化的最直接反应，就是电池的内阻变大。外在的表现就是，大电流放电的时候电流放不出来。依据以上2组数据，可以肯定故障来自于电池的老化，为此，可做如下变动：

- 在C组的电池中，剔除编号为C1的电池，换一个同规格的新电池。C1电池老化得太厉害了，把它扔掉。

- 在 A 组中,电池 A3 的性能最好。而 B 组老化的程度差别不大。用电池 A3 取代 B 组中性能最差的 B2,组成一个新的 B 组。

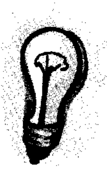
表 5.3

电池分组	电池编号	开路电压/V	短路电流/A
1500 mAh 品新电池	A1	1.490	2.46
	A2	1.479	1.90
	A3	1.480	3.60
	A4	1.491	1.16
1300 mAh 的 GP 电池	B1	1.438	3.64
	B2	1.435	3.25
	B3	1.447	3.31
	B4	1.431	3.70
500 mAh 的镍镉电池	C1	1.423	1.05
	C2	1.394	3.98
	C3	1.390	4.21
	C4	1.391	4.02

A 组、B 组中淘汰下来的 4 个电池不再用于数码相机,把它们编组起来,放在小孩子的遥控赛车上,因为赛车对电源的要求不高,所以效果也相当不错。至少也比街上 5 毛钱一个的电池好无数倍。新的 C 组电池用在照相机上,再也没有出现自动关机了。B 组电池用在照相机上,虽然偶尔也会自动关机,但是发生的频率有明显下降。

电池老化是电池使用中的自然规律,同一批次的电池,老化的速度都可能不同。老化的电池在大电流放电的时候,其端电压会迅速下降。你可以回过头看看前面的故障描述,以上故障发生概率高的时候,基本都是在照相机的工作电流非常大的时候(比如开闪光灯、开照相机的后置取景器),大电流放电的时候,老化的电池首先电压下降。而一组电池中只要一个电池的电压很低,则照相机内部的电源管理电路就认为这一组的电池放电放光了,进而自动关机。

所以通过这样的数据采集和分析,将老化的电池剔除,剩下的电池按照其放电特性重组,就能延长电池组的使用时间。不至于因为一个电池的老化而处置一组电池。



- 一次只看一个两个数据，不一定能看出什么东西。但是一次看一大把数据，加上一定的数据深度分析，经常就能推断出问题所在。这个区别，犹如站在山脚看一个城市和站在山巅看一个城市的差别。
- 统计数据，是一个很耗费时间、重复性很强的工作。很多人对此不会有耐心。但是处理一些比较难搞定的故障，这些数据却非常有用。

5.4　找不着北的时候怎么办？

以上说了这么多方法，虽说都是查找故障的方法，但是这些方法基本都是有脉络可循的时候才能使用。但是在研发设计过程中，经常会出现一些说不出什么道理的故障，或者不知道顺着哪个方向才能找得出来的问题，而且这样的问题还不在少数。

当你找不着北的时候，最要紧的就是充分利用前面所陈述的“望、闻、问、切、诱”，尽量多、尽量细地收集和故障相关的信息。望、闻、问、切是维修判断过程中的第一要法，它贯穿于整个维修过程中。观察不仅要认真，而且要全面。要观察的内容包括：

- 故障发生时候的周围环境。
- 硬件环境，包括接电压、电流、线缆连接状态等。
- 哪怕是很细小的，和平常不一样的东西和迹象，就像警察在杀人现场不放过一根头发丝一样。
- 积极去了解用户操作的习惯和流程。这些习惯和流程经常和设计者的习惯和流程不一样，所以一些故障不会在设计者手上体现出来，而只会在对用户一侧才能体现。

有了以上这些信息，多半不会再找不着北了，至少也会将故障范围缩小一些。

如果故障范围缩小后，仍然不能具体判断问题所在，除了以上的那些招式外，还可以试试最小系统法，主要是要先判断在最基本的软、硬件环境中，系统是否可正常工作。使用这个方法的时候，一般都是很茫然的时候，这个时候请首先要相信每个模块都有出问题的可能，将最小系统以外的东西尽可能地用大砍刀

砍掉。最小系统建立好了之后，如果不能正常工作，即可判定最基本的软、硬件部件有故障，从而起到故障隔离的作用。

如果这个时候故障消失了，说明故障点在砍掉的那些部分里面。这个时候就需要用到逐步添加法。最小系统法与逐步添加法相结合，能较快速地定位发生在其他模块的故障，提高查找故障的效率。

逐步添加法，以最小系统为基础，每次只向系统添加一个设备或软件模块，来检查故障现象是否消失或发生变化，以此来判断并定位故障部位。逐步去除法，正好与逐步添加法操作相反。逐步添加/去除法一般要与替换法配合，才能较为准确地定位故障部位。

我以前在做一个软件的时候，就碰到过这样找不着北的情况。这个软件工程的层次结构如图 5.22 所示。现在我们怀疑 A 模块出问题了，就可以将软件工程进行裁剪，只留下硬件层和模块 A，如图 5.23 所示。但是这种情况下，软件可以正常运行。于是将剩下的模块逐个顺序加入，加入模块 D 后问题出现，而这个故障的外在现象却很像模块 A 的故障。于是将目光从模块 A 转移至模块 D，很快就找到了故障。

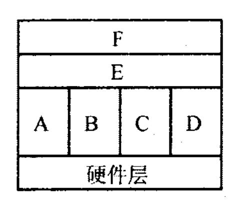


图 5.22

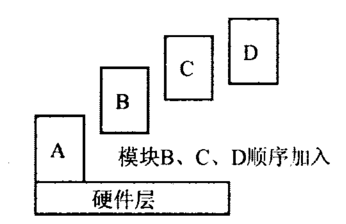


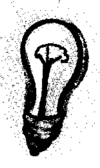
图 5.23

在设计中有很多故障，根源在 A 处，故障现象却体现在 B 处、C 处和 D 处。就好像痛风患者关节疼痛，虽是关节的故障，但根源是来自于不良的饮食习惯；糖尿病患者伤口很难愈合，不是因为血小板水平低下，而是源于微循环系统的病变一样。这样的故障因为现象和实质的关联度较低，理解起来也很困难，而且查找难度极大，而且出现的频率也不算少。比如电源纹波会引起某个特定硬件模块的故障；比如电视机中高频、中频部分设计不良，会导致屏幕的画面有条纹、图像不清晰等。刚着手解决这些故障的时候，都会让人摸不着头脑。这就需要用前述的种种手段来发现问题所在。









提示：

- 查找故障的过程，归根结底都是收集信息的过程。
- 很多故障，处理的方法很简单。但是知道问题在哪里却很难。所以找故障比处理故障重要得多。
- 综合运用以上的方法，学会怎么查找故障，对于您提高设计水平，大有好处。



6

无电路图维修的技巧

首先重申一个定律：20－80定律。这个定律的大意是说：少数20％的东西，占据了80％的另外一种东西。比如这个世界大约20％的人占据了大约80％的财富；地球上20％的国家占据了80％的石油资源；太阳系中80％的质量集中在20％的天体上面；20％的疾病是80％病死者的直接死亡原因，等等。

做电子工程师这么多年，经常会有人拿一些杂七杂八的东西来维修。而且通常除了实物以外，没有任何书面资料，更没有维修时最需要的电路图。不过这些东西做久了，发现所有的电子电路，80％的故障也是集中在20％的部件上，为此罗列出我的一些经验：

首先需要说明的是：没有电路图而要维修一个电路板，犹如一个医生蒙着眼睛对病人治病一样。能够治疗好，要有许多经验和运气的成分，但是和做其他事情一样，只要能够尽量多的收集信息，再根据工程经验的积累，把东西修好的概率还是有的。尤其是一些复杂度不高的设备。

一般拿来维修的每种电子产品，基本都有其故障高发的点。一般而言，一个产品中各个模块的故障率都不尽相同，就如同本书中《电子元件故障发生概率排行榜》一章中所叙述的那样。

下面以表6.1所列的形式来分析各种设备内部的故障高发模块。（注意，是模块，不是元件！）

表 6.1

设备种类	容易损坏的部分	损坏的原因
任何设备	会发热的部分	发热元件的焊点，在机器使用多年之后，由于反复的发热、冷却而热胀冷缩，导致焊点开裂。所以碰见有发热部件，又使用了好几年的电子设备，比如CRT电视机和监视器。不用动脑子，先把这些会发热的焊点仔仔细细、一个不漏的全部焊接一遍，至少有30％～50％修好的概率

续表 6.1

设备种类	容易损坏的部分	损坏的原因
任何设备	电源部分	无论什么设备，这个部分都是故障高发部分。 • 如果是开关电源，一般都是开关管被击穿了。通常伴随保险丝熔断 • 开关电源的启动电阻，一般是高阻值、大功率的电阻，也是故障的高发元件 • 如果是变压器的电源，一般是超载导致稳压部分损坏或者是变压器过载烧毁
	模拟信号部分	这个部分发生故障的概率要低很多，一般体现在以下几种情况： • 电解电容干涸 • 瓷介电容漏电 • 晶体停振或晶体滤波器失效 • 大阻值电阻开路或者阻值严重偏差 • 其他故障
	CPU、CPLD 和 FPGA 部分	这个部分发生故障的概率不太高，一般就是以下 2 种情况： • 负责 CPU 主频的那个晶体停振 • 一些设计不良的产品（一般这样的产品产量也不高）会出现 flash 中的代码丢失，重新刷新 flash 后一般可以修好
	除了 CPU、CPLD 和 FPGA 部分以外的数字电路	这个部分发生故障的概率很低，可以说很不常见。如果有损坏的，基本上都是被静电或者强电打坏的
	板间连接器或者连接线	这个部分的故障率非常高，尤其是以下这些设备： • 廉价设备。不用说了，便宜没好货。便宜的机器，它的连接器也好不到哪去 • 使用了很久的设备 • 在高湿度、高盐度及腐蚀性气体环境下使用的设备
CRT 电视机和监视器	扫描部分	一般是高压包老化击穿或者行管击穿，经常导致电源模块继发性故障
LCD 电视机和监视器	LCD 屏	一般都是机器经过暴力冲击后损坏的，有时候是小区域的，有时候是大片的损坏。这个时候除了换屏，其他方法都无效

续表 6.1

设备种类	容易损坏的部分	损坏的原因
数码相机和手机	电池	这是最常见的故障。尤其是使用了一年半载的设备。电池如果不良的话,老化的速度很快。最直接的故障是设备工作时间下降。电池损坏有时候还会带来非常匪夷所思的故障,比如突然自动关机。一般自动关机都发生在设备耗电比较大的时候,比如手机通话的时候、照相机在拍摄的时候及在给闪光灯充电的时候等。而且电池老化越严重,自动关机的概率越高
	LCD 屏	一般都是机器经过暴力冲击后损坏的,有时候是小区域的,有时候是大片的损坏。这个时候除了换屏,其他方法都无效
VCD 和 DVD	激光头	一般是激光头老化,能换就换一个。不能换就再买一个整机
	信号电路板	一般这个部分不会坏,要是真的坏了,一般也不会有电路图供你维修

好了,有了以上提供的信息,至少已经可以搞定 30%的故障了。那剩下的问题一般就是没有资料导致的。如何得到更多的资料呢?看看下面的例子:

我曾经修理过一个计算机的 ATX 电源,没有电路图。拆开后,其中的主芯片是 TL494,没有电路图,就去网上找一份官方的 IC 规格书来,那里面详尽描述了 IC 各个 pin 的功能。真实的电路和数据手册(datasheet)上的典型应用电路一般不会有重大差别(见图 6.1 和图 6.2)。

有了规格书,我就知道这个 IC 有一个振荡脚,是第 5 pin 和第 6 pin。拿示波器,观察这里,没有看见任何波形。于是拿万用表测量该 IC 的供电脚第 12 pin,发现没有电压。原来是核心 IC 没有供电导致整个电源不工作。进一步追踪,发现电源启动电阻的阻值变成了无穷大。用更大功率的同阻值电阻替换后,故障解决。

类似的手段,还可以处理很多使用通用芯片设计的电路板。比如收音机及电视的信号部分等。掌握了这个办法,机器被修好的概率应该可以增加 10%。

如果找到了损坏的器件,但是没有备件更换怎么办?

对于这个问题,要分开来解决。如果是像三极管或二极管那样的简单器件,通过对比规格书,可以很容易找到代换品。但是如果碰到特殊 IC,有时候就只能放弃了。因为小批量采购是很麻烦的,有时候买一个器件付出的比挣的维修


感悟设计



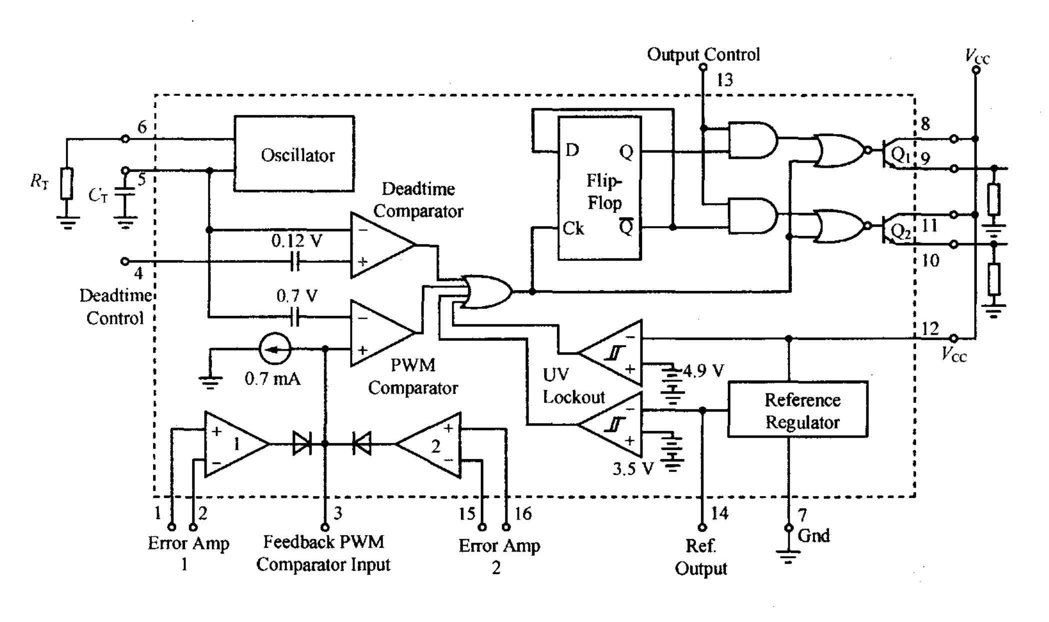

Output Control
13
V_{CC}
6
5
R_T
C_T
Oscillator
Deadtime
Comparator
D
Q
Flip-
Flop
Ck
Q̄
Q_1
Q_2
8
9
11
10
12
0.12 V
4
Deadtime
Control
0.7 V
0.7 mA
PWM
Comparator
UV
Lockout
4.9 V
3.5 V
Reference
Regulator
V_{CC}
1
2
3
15
16
14
7
Error Amp
1
Feedback PWM
Comparator Input
Error Amp
2
Ref.
Output
Gnd

图 6.1

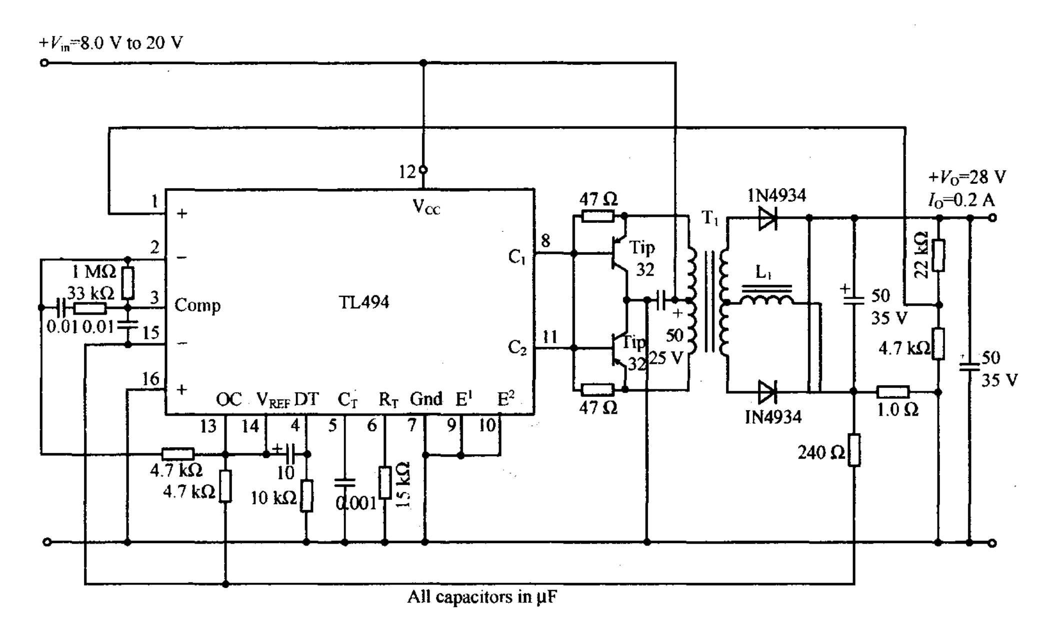

$+V_{in}$=8.0 V to 20 V
12
1
2
3
15
16
Vcc
Comp
TL494
C_1
C_2
8
11
OC
V_{REF}
DT
C_T
R_T
Gnd
E^1
E^2
13
14
4
5
6
7
9
10
1 MΩ
33 kΩ
0.01
0.01
4.7 kΩ
4.7 kΩ
10
10 kΩ
0.001
15 kΩ
47 Ω
47 Ω
Tip
32
Tip
32
50
25 V
T_1
1N4934
1N4934
L_1
50
35 V
50
35 V
22 kΩ
4.7 kΩ
1.0 Ω
240 Ω
$+V_O$=28 V
I_O=0.2 A
All capacitors in μF

图 6.2

费还多。如果实在找不到替换的器件，可以考虑更换整个模块。

以前碰到一个纺织机上的控制电路板，把220 V开关电源、控制部分和电动机驱动部分都放在一个双面板上。最后电源部分坏了，还不是那种常见的比较好搞定的电源故障。经过分析，发现这个电源只有+24 V、5 V和-12 V三组电源输出。最后为了节约维修时间，按照这样的规格去买了一个开关电源模块，用电线跳接到电路板上，就修好了。这个例子说明：如果维修所投入的时间按照收入折合成为货币，这些货币超过买一个功能模块的代换品的话，那就应该去买一个这样的功能模块替换。省下时间就是延长了你的生命！

哪怕替换的东西差别很大，只要替换的东西功能一样，而且没有重大副作用，就可以替换。为了说明替换法维修，再举一个和电子无关的维修例子：

一次出差以特惠价买了一台遥控直升机，该直升机有上下两层旋翼。直升机上面有一个平衡杆以稳定上面的旋翼。因为旋翼的迎风角度如果不正确的话，那这个遥控直升机的操纵性就会很差，所以一般的玩具直升机上，都有一对真正直升机上没有的平衡锤（见图6.3），这样，当平衡锤被高速旋转后，可以使平衡锤的旋转平面严格垂直于引擎的旋转轴，而顶上的旋翼也因此被校正到严格垂直于引擎的旋转轴，以保证整机的易操作性。

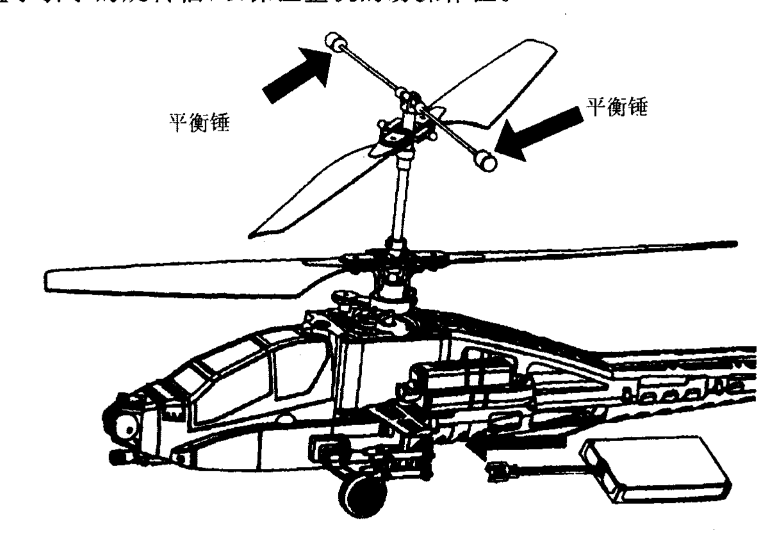


图6.3

这个平衡杆的两端是2个金属配重块，如图6.4所示，它们和平衡杆之间用螺纹连接。但是这个飞机在飞了几天后，突然发现机身剧烈震颤。检查后，发现

原来 2 个金属配重块在飞行的时候被甩掉了一个。而这个部件根本买不到。想了很多办法去试图找到这样一个重量差不多的金属块，但是即使找到了，也无法和平衡杆进行可靠的固定。最后经过很久的尝试和思考，找到一个替代物，就是：铅笔。

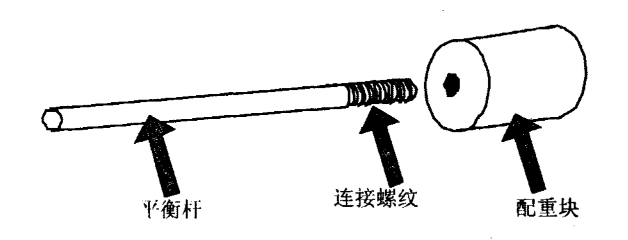


图 6.4

把一根新的铅笔，从中间截断，变成长度和重量相等的 2 部分。这样半根铅笔有和原先金属配重块大致相等的重量，只要把铅笔用有弹性的电工胶布紧紧地缠在平衡杆的两端(见图 6.5)，且两个铅笔固定的位置精确相等，那么这个平衡杆的动平衡就能得到保证。这样更改之后，直升机一样能飞。虽说用铅笔改过的平衡杆的迎风面积增加了不少，会导致旋翼的飞行阻力增大。但是无论如何，通过这样修理，飞机总算是能飞了。

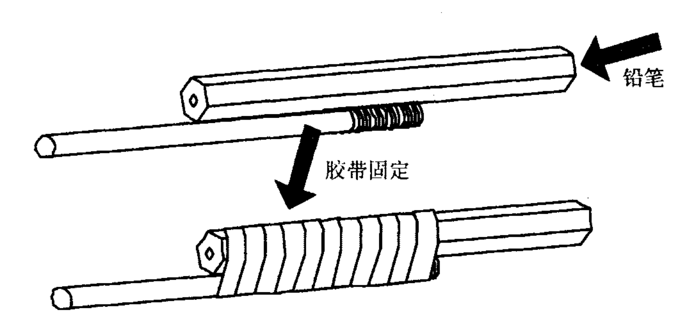


图 6.5

这样更改后，铅笔可以胜任的理由如下：

- 铅笔的形状和尺寸虽然差异很大，但是它们可以完成相同的功能。
- 半根铅笔有和原先金属配重块大致相等或者更重的重量。
- 铅笔比较长，比原先的金属配重块更长。这样固定后，平衡杆的总长度更长，会更加平衡一些。这个道理就好像杂技演员走钢丝的时候要在手里拿一根很长的杆子一样。

- 因为半根铅笔的长度比较长，这样通过胶带固定后，可以和下面的金属平衡杆有更长的接触面，这样被胶带紧绷后，就会有足够的摩擦力，就可以保证飞行的时候，铅笔不至于被旋转的强大离心力甩出来。因为没有螺纹连接的话，就只有指望摩擦力来保证旋转时的牢固性了。

就在写本文的前几天，我的一个 NOKIA 的智能手机，因为摔在地上，发生故障，现象如下：

1. 无论打入、打出电话都没有对方讲话的声音。
2. 打入电话的时候会有铃声。
3. 如果打入、打出电话的时候，切换到免提模式，一切正常。
4. 可以正常播放 MP3、MP4 等多媒体文件。
5. 平时的使用经验告诉我：这个手机有一前一后 2 个扬声器，凡是大点的发音都用后面的扬声器，小点的发音都用前面的小扬声器。

综合以上几条，可以推断出，因为第 3 条的缘故，手机的信号处理部分发生故障的概率不大，第 2 条和第 4 条说明它们可能使用机器后面的那个大声的扬声器。也就是说手机电路板发生故障的概率不大，最有可能的是手机前面的那个小扬声器开路了。而根据我的感觉，扬声器发生开路的概率要高过电路板损坏的概率。按照这个思路，小心翼翼的拆开手机，发现果然符合我的判断。花 10 元钱买了一个同样的扬声器后，手机就好了。因此节约了几百元的修理费。

还有一个例子，一次公司里面的示波器坏了。故障现象是示波器的 LCD 屏幕的背光时有时无。打电话给厂商，说是维修周期要 15 天。因为我以前设计过 LCD 电视，所以对 LCD 的背光略知一二。LCD 的背光控制电路归根结底是一个高压振荡器，电路本身并不复杂，所以类推这个示波器的故障，也应该不会很难处理，估计自己修理电路板可能会有胜算。拆开机器后发现机器里面只有 2 个大电路板，一个是电源板，另一个是信号处理的电路板。而 LCD 背光灯驱动电路就在电源板上。背光灯驱动电路在电路板上的位置可以轻松找到，因为 LCD 上有一个 2pin 的连接器连接到这个电路板上。又因为这个示波器的使用已经超过 3 年了，我估计和大多数彩显和电视机的故障一样，是因为发热导致焊点开裂。把这个电路板放在强光灯下对这个电路板的焊点逐个进行外观检查，果真在高压变压器边上发现一个开裂的焊点，重新焊接后示波器马上恢复了正常。

有了以上几条原则和对故障的推理方法，一般的机器都至少有 50%～70% 的概率能够较快修好。当然，不可否认，如果电子设备故障出现在电路的核心部

分和信号部分,那没有电路图真的很难修理。比如现在越来越多的精密的数码产品等。罗列的那些故障,通常就要花费很多时间去琢磨,这样就得不偿失了。这个时候,应遵守经济学的规律,赔本买卖还是不做为妙。不过好在这样的故障也符合 20－80 定律,只占大约 20％～30％。



7

设计中的概率论

7.1 无处不在的概率

以前看过一个电视剧，讲八路军和日军打仗的故事。其中一个战例是：

一次八路军得到情报说：有一个营的日军会长途奔袭到某地，途中会经过一个山地。于是八路军伪装后埋伏在其中一个山谷两侧，想一口吃掉这块大肥肉。这个山谷两边都是山，中间是一条小河和土路，属于典型的易守难攻的地形。而日军军官也明白在山区行军的风险，所以做了以下决策：

- 把日军分成几十个小队，每个小队只有十几个人。
- 每十五分钟才出发一个小队。
- 每个小队的日军士兵之间至少保持 20 米的距离。
- 在每个小队前面 300 米始终有一个侦查兵，每经过一个路口的转弯，或者别的危险路段，就拿望远镜先观察前方是否有危险，确认安全后，用旗语告知后面的小队可以前进。

这样一来，八路军就遇到麻烦了。虽然八路军隐蔽得相当不错，始终没有被日军发现。但是每次只有十几个日军从眼皮底下经过，消灭他们吧，十几个人还不够八路军塞牙缝的，而且开火后，前面后面的日军都会施援。不消灭他们吧，却不断的有一队队的日军经过。左右权衡下，为了不陷于被动，最后八路军不得不取消了这次行动。

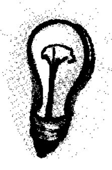
在这个战例中，日军极好地控制了风险。首先，日军明白在山区行军，中埋伏的概率很大。而且一旦被伏击，中国军队居高临下，日军会死伤无数。所以日军在每个小队前面 300 米安排一个侦查兵，可以预先侦查是否有埋伏。这个措施就降低了进入伏击圈的概率（由于八路军伪装得很好，骗过了所有的侦查兵，所以这个对策算是失效的）。除此之外，日军把自己分割成几十个小队，每个小队只有十几个人，每十五分钟才出发一个小队。这样就把整个行军的队形拉得

很长，可以降低整个部队被围而歼之的概率，万一真的被袭击，也不至于有重大损失，各个小队也可以相互策应。每个小队的日军个体之间至少保持20米的距离，说明它的每个小队的队形也拉得很长，可以降低被一个炸弹一次干掉好几个的概率，也可以降低被机枪集中射击的概率。

通过以上一环扣一环的措施，日军毫发无伤地穿越了对自己不利的地形，由于充分分析了各种风险因素，而且对每种风险因素都有可操作的减小损失的具体措施，所以他们从八路军的枪口底下走过，八路军也不敢贸然开枪。这就是因为日军充分估计了各种风险，并且拿出了相应的对策。他们的每一个对策都可以降低被进攻的概率，几个对策加在一起，便有了非常高的安全概率。

再举一个例子：前几年经常有煤矿出事，今天这个瓦斯爆炸了，明天那个矿井漏水了。都有死人，而且经常一下死好多人。其实这些事故的原因统统只有一个：忽视风险。可能那些煤矿老板会说："中国有那么多煤矿，一年出事在电视上报道的也就那么几个，我不会这么衰的"。而且为矿井增加安全设施，通常就意味着他们眼前的金币立刻要少一大堆。一面是安全资金的投入导致现金的立刻减少，一面是发生概率不足千分之一、万分之一的事故，很多人都会选择把现金留下。当他们美滋滋地点着一堆堆钞票，切实感受到传说中的"数钱数到手抽筋"的时候。身后的矿井却轰然倒塌。那时他才发现，人死太多了，矿工的家属不会放过他，政府更不会放过他，这个时候拿出所有的钱也摆不平这个烂摊子。

这样的例子说明，我们做事的时候，会有一些问题，发生的概率很小，但是体现的问题却很大，用数学语言精确描述，就是："小概率、大权重"。这些人由于只看见了小概率，却没有想到这个小概率绑定的却是大权重，所以到头来那些黑心矿主的金币统统变成了学习概率论的学费。



这几个故事告诉大家：

- 任何时候，都不要指望会有好运气。因为这个世界大家都是吃五谷杂粮的，这个世界没有神仙。
- 请一定按照事物的客观规律办事。否则今天不出事，算你走运；明天不出事，也算你走运。但是总有一天风险要连本带利兑现给你。
- "小概率、大权重"的事情，应该把它当作大概率事件看待和处理。

- 在设计过程中，要千万注意每一个可能产生风险的细节。
- 要估算各个环节中产生风险的概率，及风险产生后的损失。根据这个安排防范措施。
- 不要指望一招可以通吃天下，安全源自一大堆细节的完美。

Ok，故事讲完了，这些故事对做设计有什么用呢？第一个案例，说明要注意设计中的每一个细节，每个细节都保证出问题的概率很小的话，那整体出问题的概率就很小。第二个案例说明，很多事情和设计一样，要把产品做好，需要把握好每一个细节。而只要忽略掉一个细节就可能会满盘皆输。所以尤其要注意那些会有"一票否决"可能的风险发生点。

7.1.1 注意设计中"一票否决"的地方

以下地方都是足以一票否决设计的缺陷，这些都是"小概率、大权重"的事情。需要高度关注：

- 整机设计，功能缺乏新意，客户没有购买欲。
- 电源的失效。
- 复位电路的失效。
- CPU 主振荡器的失效。
- 电路不能耐受 ESD 打击，被打击后电路产生不可逆的损伤。
- 软件死机后不能自动恢复正常。
- 输入单元的失效，比如键盘的失效。
- 输出单元的失效，比如 LCD 显示器的失效。显示的内容有多余的东西，比如条纹、花斑等。输出的信号有杂音等。
- 软件指针的错乱，对一个空指针进行操作，或者指针指向的地方是一个已经释放掉的内存块。有这样一个错误就足以引起 CPU 死机。
- 软件在特定情况下会陷入死循环，造成 CPU 死锁。
- 足以引起用户感觉异常的故障，这种情况就非常多了，而且故障现象形形色色，不胜枚举。

7.2 想好了，出手以后有多大的胜算

前几年有一个轰动性的新闻，几个农业银行金库的职员利用职务之便，盗窃

5100 万现金购买彩票。我看了这个新闻的深度报道，第一感觉就是：这些银行职员真是胆大妄为；第二个感觉就是：这些银行职员真的很没有文化。这里暂且撇开他们拿国家的钱去买彩票，假设他们拿自己的 1000 万元去买彩票会如何？

首先拿自己的 1000 万元去买彩票，无论输赢，都不用承担法律责任，说不定彩票发行机构还会给这样的大款发一面锦旗或者牌匾之类的，表彰一下这样的阔佬。但用自己的钱去买彩票就没有问题吗？

这里，他们或许是忽略了、或许是压根就没有明白这样一个事实：彩票发行机构本质上是一个需要盈利的机构，而它的盈利是建立在价值转移的基础上的，而不是像银行、工厂那样建立在价值创造基础上的。所以它们的盈利是来自广大彩民的买彩经费，虽然彩民也能经常中一些小奖，偶尔还会有一些运气好的人真的可以拿走五百万的大奖，但广大彩民的买彩经费总会有一部分成为彩票发行机构的利润，还有一部分用于彩票发行机构的运转。我在一个财经网站上看见了这样一个额度的百分比：在中国，彩票发行机构返还给彩民的额度平均是 70%。彩票机构的利润和发行成本占 30%。中奖以后政府还要以税收方式拿去 20%。

基于以上分析，可以得出以下结论：

如果是拿小钱去小赌，返还概率是 70%，这样的返还要么是一个巨奖；要么就是一段时间的连续购买，连续 0 收益。一下一个巨奖的概率自然是极低的，否则也不会成为新闻。而一段时间的连续购买，连续 0 收益或者负收益却是大概率事件，经常伴随那些好赌的彩民一生。

数学《概率论》里面有一个定律叫"大数定律"。这个定律描述如下：

在随机事件的大量重复出现中，往往呈现几乎必然的规律，这个规律就是大数定律。通俗地说，这个定理就是，在试验不变的条件下，重复试验多次，随机事件的频率近似于它的概率。比如，我们向上抛一枚硬币，硬币落下后哪一面朝上本来是偶然的，但当我们上抛硬币的次数足够多后，达到上万次甚至几十万几百万次以后，我们就会发现，硬币每一面向上的次数约占总次数的二分之一。偶然中包含着必然。

所以，买彩票的时候，无论少量购买还是斥巨资购买，从宏观的角度看，其投资收益率都是固定的，符合这个大数定律。也就是说：一个大额彩票买家的投资收益率和一大堆小额散户彩票买家的投资收益率是基本一致的。小额散户彩票买家虽然中奖的概率小，但小赌损失也小，绝大多数人都能承受。大额的彩票买家则不同，大额资金购买彩票，虽然一次中奖的概率大多了，但是一次的本钱

也大多了，会产生大额亏损。而且金额越大，亏损的百分比越接近固定的常数。亏损率就是100%减去庄家设定的返还率。但是这几个银行职员并不懂这个概率论常识，把小赌的以小博大，简单的类推为以大博大。结果就会身不由己地越赌越输，越输越赌。这些银行职员大概原本是想以大博大，把赢来的钱补上银行的窟窿就可以了。但是由于他们数学知识的缺失，对自己出手后的胜算产生了错误的评判，最后的结果就是：他们需要用自己的生命为这种错误的评判买单。

《三国演义》中有很大篇幅讲诸葛孔明出祁山攻打魏国的故事。诸葛孔明攻打魏国，因为劳师远征且蜀道艰险，后勤保障成本高。所以一向都是以"一鼓作气，再而衰，三而竭"为其指导思想。因为速战速决才能有效地降低战争成本。所以对诸葛孔明而言，他取得最大胜算的战争方式就是速战速决。而对于诸葛孔明的对手司马懿而言，他在本土作战，熟悉地理人文，而且其战场的后方是大平原，后勤保障成本比蜀国低得多，而且魏国的综合国力也远高过蜀国。所以对于司马懿而言，只要避过敌人的锋芒，打持久战才是他的最大胜算。司马懿和诸葛孔明这两个项目经理在明白各自的胜算后，都绞尽脑汁地贯彻自己的方针。诸葛孔明是想方设法、绞尽脑汁诱使对方出战，而司马懿却是宁可穿上诸葛孔明送来的女人衣服也坚守不出。在这2种思维的博弈中，诸葛孔明速战速决的想法始终不能实现。而他受客观条件的限制，也无法改变战略进行持久战。所以即使他先后六出祁山也始终没有捡到魏国的多少便宜。

有一段时间，我经常要出门坐公共汽车，去市内的某个地方办事，如图7.1所示，左边的圆圈是我的目的地，右下角的房子是我家。如果要直达的话，可以走路0.3 km后选择线路1，就可以到达目的地。可是要命的是，听说这个线路的经济效益不好，公交公司压缩了这个线路的运力。所以一般等20分钟才可能来一辆线路1的车。因此，经常迟到。比较郁闷。

我不是公交公司的调度，所以不可能知道什么时候有什么车会开到我面前。但是我可以在宏观面上对这个问题进行分析，通过对所有公交线路的比对分析，我发现线路1、3、4和5都可以到目的地，而且每个线路大约5分钟肯定会有1班，所以可以这样乘车，先从线路2，到图7.1中的换乘站，那里就有4个线路，可以到达目的地。在那个换乘站，一分钟内等到车的概率就是1/5+1/5+1/5+1/20=65%。所以我换了一个等车的方式后，虽然这样多转了一次车，但是我在单位时间内等到车的概率大大增加。所以迟到的概率也小得多了。

在设计中也是这样，无论是做软件还是做硬件，在着手之前都要思虑再三。一个硬件的原理图完成后，基本就决定了以下方面：

● 电路的抗干扰度。虽然很多人认为这个取决于PCB设计更多一些，但

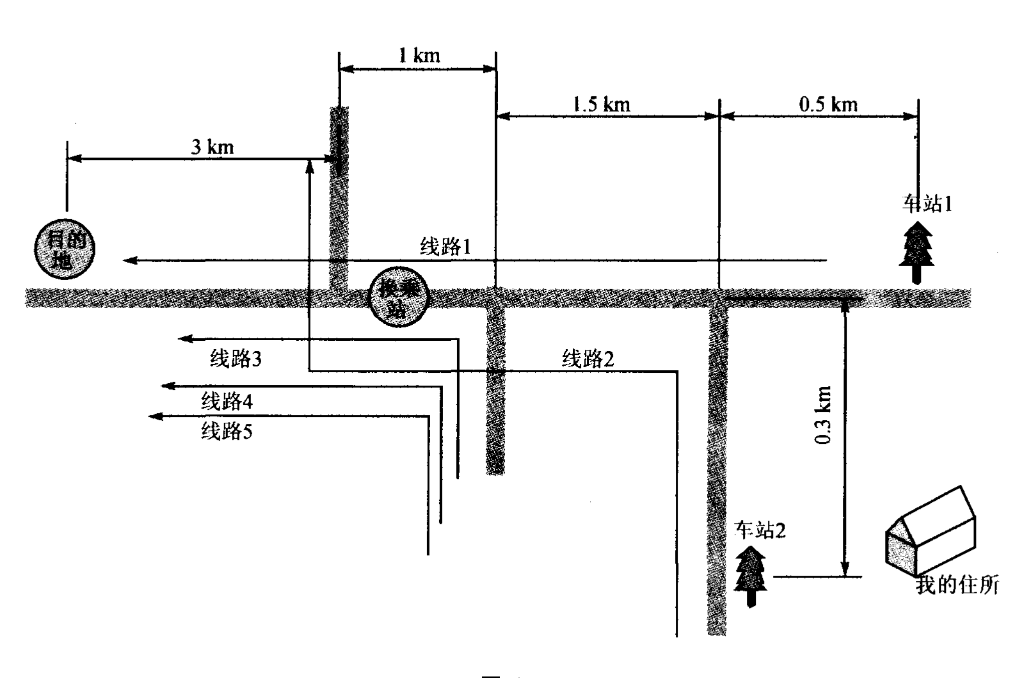


图 7.1

是根据我的看法，至少 70%硬件方面的抗干扰不良，都是由于原理图上就缺少抗干扰的措施。因为在我看来，画 PCB 虽然也很有技巧，但归根结底是对原理图设计思想的执行，而思想有错也就是原理图有错的话，思想执行得再好也是事倍功半。

- 制造成本。
- 维修成本。
- 元器件的采购方便与否。
- 抗异常环境能力。比如静电打击、电源反接及高低温导致的电路参数飘移。

所以，从画原理图开始，就要小心。

还有，做软件的时候：

- 软件的程序逻辑流程会决定用户的操作流程。一个不好的程序逻辑流程，会降低用户的操作便利性。
- 一个数据结构，可以决定一个程序怎么走。数据结构设计的好，程序也会简单很多。用户也容易理解。
- 一个算法，采用不同的设计方法，需要的内存数量、运算精度和运算速度

都会有天壤之别。

所以，兵马未动之前，一个良好的战略，其实就已经能够在很大程度上影响战争的结果。同样的，设计还没有开始前，一个良好的设计构想和策略，也能决定你设计的高下。

7.3 设计中的“大数定律”

设计中经常会碰见很多异常现象。有的异常现象是有规律的、大概率地重复出现。而有的异常出现的概率很小，经常一晃而过。不够细心的人，会认为这是偶然现象，就放过去了。这些小概率的问题，要么是由于电路上元器件的安装错误、元器件离散性等原因造成的；要么就是由于特定的外部条件触发的（比如出现概率很低的干扰、用户的一些离经叛道的操作流程和操作方法）。如果在设计的时候把这些小概率的问题放过去，这些问题就会以另外一种形式体现出来：就是在量产的时候出问题或在长时间使用的时候出问题。

7.3.1 产量的大数

产量的大数，就是所设计的机器在量产以后，产量达到几百乃至几万后，可能体现出问题来。设计的时候，如果放过一个出现概率为 1% 的故障，那量产 10000 台机器，就可能会有 100 台故障机。而要处理这 100 台故障机，需要耗费比生产 100 台机器更多的工时。这些小概率问题，大致由以下原因造成：

- 由于电路上元器件的安装错误。如果是电路上元器件的偶然性安装错误，只要 BOM 表的数据没有差错，那异常就不会继续下去。
- 设计者计算元器件参数，本来就设计得有一定的偏差。这样机器偏离了最佳的工作点，偏离得还不多，所以大部分机器能够用，有问题的只是小概率。但是这种小概率在大产量下，故障机的总数会被线性放大。
- 元器件离散性等原因造成的。这种原因比较好理解。却经常被设计者忽略。所以关键位置的元件，需要分别计算元器件的最大值和最小值。看看电路在这些最大值和最小值下，还能不能正常工作。
- 其他设计的不良，造成电路或者软件偏离了最稳定的那个点。本书的“设计的中庸之道”，其中讲电话机侧音的案例就是所谓“最稳定的那个点”的最好解释。

曾经设计过一个单片机系统，内部的 MCU 使用了一个 4 MHz 的晶体振荡器，这个晶体的规格书上写了，相配的谐振电容是 10 pF（就是图 7.2 的 C_1 和

C_2)，而设计工程师没有注意这个细小的差别，而按照 32768 Hz 的晶体设计，将 C_1 和 C_2 设计成了 27 pF。在设计阶段基本没有问题，只是在非常个别的时候，会有上电的时候 CPU 不会跑，重启一下就好。由于该问题出现的概率太低，当事者没当回事就把这个问题放过去了。但是在中试 100 台机器的时候，就有 2 台很难开机。经过检查，发现，如果开不了机，那 CPU 的所有总线信号都没有了，怀疑是晶体的问题。把晶体换一个就好了，以为这个故障就到此为止了，但是换了晶体的机器，没几天，又坏了，还是很难开机，并且 CPU 的所有的总线信号都没有了。后来校对元件才发现 C_1 和 C_2 设计错了。为了防止这些机器出厂以后再出问题，所以把这批中试 100 台机器的 C_1 和 C_2 全部更改成了晶体规格书上写的 10 pF。这个错误，造成的故障是一个发生概率较低的故障，比较难发现。但是在大产量下，就会有一定的故障数量，暴露出来。

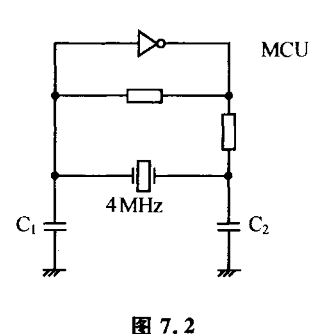


图 7.2

7.3.2 时间的大数

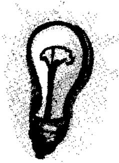
时间的大数，是说很多事情在时间的考验下，会体现出其深层次的优点和缺点，甚至根本性地改变一个事物的基本特征。像"路遥知马力，日久见人心"、"水滴石穿"和"铁杵磨成针"说的都是这个道理。

众所周知，老年人患病的概率大大高过年轻人。而老年人患的病，多是一些需要慢性积累的病。比如糖尿病、心血管病和痛风等。这些病，普遍都是难以治疗，现在的医学科技(包括中医)都只能治标不能治本。这些疾病发展到后期，基本都会要命(即大权重故障)。这些疾病在年轻人身上发生的概率很低。而且年纪越小，发生的概率越低。这些疾病，多是由于患者的不良生活习惯，经过多年慢性积累造成的。这就是时间的魔力。而一旦这些疾病的症状体现出来，再要回到原始的状态就很难了。

同样的，要看一个人的身体好不好，光看一时不行。因为有可能一些人会经过化妆和粉饰。看上去气色很好的人，其实在脂粉之下有可能掩藏着大毛病。而这样的人，只要连续观察几天，大多能得出很准确的判断。因为时间一长，你观察到的信息也会多很多，他身上的问题暴露的概率也会大幅度上升。这个道理类似于"路遥知马力，日久见人心"。

时间的大数，在考验软件长期运行能力的时候，是测试其品质的重要方法。

会有死机隐患的软件和硬件，在大数量的时间考验下，漏网的概率很小。而一些机器的硬件，在产品线上生产了之后，也要经过几天的老化。一些军工和航天产品甚至需要几个月的老化。而这些产品如果经过了 3 个月的老化，那无故障工作到 3～5 年的概率就会极高。



提示：

- 小小的设计错误，会在量产的时候，故障概率会被放大到一个大数目。
- 小小的设计错误，会在机器长时间运转的时候，原来小权重的故障会被放大到一个可观的权重。
- 对于小问题，一定要睁大眼睛仔细盯着。因为“千里之堤，溃于蚁穴”。

7.4 勿以恶小而为之，勿以善小而不为

7.4.1 小善＋小善＋…小善＝胜利！

三国诸葛亮文曰：“勿以恶小而为之，勿以善小而不为”。这句话讲的是做人的道理，只要是“恶”，即使是小恶也不做；只要是善，即使是小善也要做。失之小节，经常是酿成大错的开始。其实在这一点上，做设计和做人也是一样。因为本章讲的是设计的方式和方法，不涉及人与人之间的问题，所以这里的“善”指的是好的事情和方法，而不是善良；这里的“恶”指的是不好的事情和方法，而不是邪恶。前面讲的八路军和日军作战的故事，就是一个将“小善”积少成多，最后克服风险，达到绝对安全的故事。

7.4.2 小恶＋小恶＋…小恶＝失败！

恐龙灭绝的原因，有一种说法是这样的：当时曾有一颗直径 7～10 km 的小行星坠落在地球表面，引起一场大爆炸，把大量的尘埃抛入大气层，形成遮天蔽日的尘雾，导致植物的光合作用暂时停止，恐龙因此而灭绝了。不过现在还有一种说法是：我们国家的科研人员，在对云贵、四川地区的地形和地下沉积层分析后，指出，恐龙不是一下子灭绝的，而是经历了 20 万年的时间，一点一点灭绝的。可能是远古时代的某一段时间，地球上的地壳活动非常剧烈，经常有火山喷发，把地层深处的物质带到了大气层，再从大气层落到了地表。其中就有一种叫做“铱”的金属元素，通过大气层和尘埃的传递，被地球上的植物广泛吸收。而恐龙

食用了这种富含铱的植物后，其生育能力一点一点地下降。虽然下降的速度很慢，但是随着时间的推移，植物体内铱的浓度越来越高，恐龙的生育能力也越来越低，最后导致灭绝。

可以计算一下，假设在20万年的时间里，恐龙可以繁殖10,000代(假设恐龙都比较晚婚，它们平均20年繁殖一代)。每代恐龙的繁殖力因为食物的原因，下降1/10000(这个数字很小了吧!)，那累计10,000代下降了多少呢？是下面这个数：

$$(1-0.0001)^{10000}=0.3678610464$$

即繁殖10,000代后，恐龙的平均生育能力只有20万年前的36%，所以它必然灭亡。因为"铱"元素的升高，虽然生育能力一次只下降1/10000，一个非常小的数字，但是经过大数量的积累，仍然会得到一个很要命的数字。

7.4.3 设计中的小善和小恶

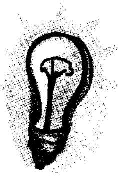
有一句名言："成功是一系列好习惯的集合；失败是一系列坏习惯的集合"。同样的，要设计出来一个稳定的产品，也需要艺术家的匠心和细致，把握好每一个细节。将每个细节出问题的概率都压缩到最低点。

1. 硬件设计中的小恶

- 对可能造成干扰的地方，重视不足，等发现干扰出现，要更改的时候就要付出高成本。
- 设计资料不同步，和别人协调工作的时候，出差错的概率大。
- 更改硬件后，不及时更新设计文档，造成改正的错误多次出现。
- 硬件模块化很差，各处的状态交叉影响。
- ……

2. 软件设计中的小恶

- 滥用全局变量。
- 程序模块化很差。
- 程序无注释，或者注释很少。
- 多个软件模块写在一个源文件里面，造成阅读困难。
- ……



提示：

- 做设计，要时时刻刻关注设计过程中的每一个异常，对异常现象不要有任何侥幸心理。
- 及时解决掉每一个细小的毛病。
- 只要是对设计有益的方式方法，可以一点点的去尝试。只要不返工，加上时间的积累，设计就能完成。

7.5 别把东西设计在临界点上

从前使用MC34063设计一个电子设备的电源，使用18 V供电，这个电源需要将18 V的输入转换到9 V，图7.3中箭头所指的A点的肖特基二极管，使用的型号为SS12。根据SS12的规格书，它的耐压是20 V，认为这样的设计应该没有什么问题。但是做出来的一些工程样机，偶尔就有烧掉击穿这个二极管的，概率大约为3%～5%。后来看了SS系列肖特基二极管的规格书，发现可以用

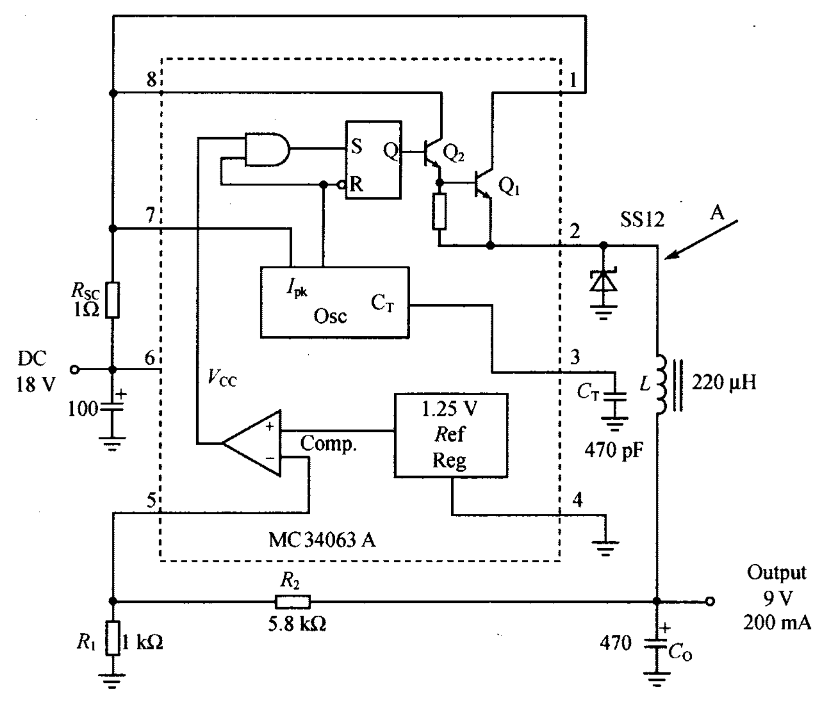


图7.3

SS14(参阅图 7.4),替换上去后,就再没有这样的问题了。

在正常工作的时候,使用 SS12,当 DC18 V 输入时,用示波器在图 7.3 中箭头所指的 A 点测量,可以看见方波。测量到方波的最高电压不过是 18.9 V。DC18 V 也是采购自专业厂家的 AC/DC 电源,电压很稳定。至少从规格书上看,这样的设计,不应该有问题。

	SYMBOLS	SS12	SS13	SS14	SS15	SS16	UNITS
Maximum Recurrent Peak Reverse Voltage	V_{RRM}	20	30	40	50	60	Volts
Maximum RMS Voltage	V_{RMS}	14	21	28	35	42	Volts
Maximum DC Blocking Voltage	V_{DC}	20	30	40	50	60	Volts

图 7.4

而我是这样看这个问题的:

- 无论是 SS12、SS14 二极管,还是别的什么电容、三极管之类的,它们在规格书上承诺的耐压,都是它们合格品的最低耐压。
- 这些器件真的拿去做破坏性试验的话,一般耐压会比规格书上高出至少 20%。因为半导体工厂生产这些器件都要保留一些安全余量。这就是为什么大部分机器不会出问题的原因。
- 如果工厂提供的器件没有比规格书高出 20%,而是刚好达到规格书的最低要求,那也不能说这个器件有问题。因为也算是达标了。
- 在图 7.3 中的电路里面,如果 DC18 V 的供电电压有一点上偏差,如果 A 点发生了轻微寄生震荡(这个在 DC/DC 里面经常会有),那 A 点的电压就会小幅度越过 20 V。
- 如果 A 点的电压小幅度越过 20 V,又刚好安装了一个没有比规格书高出 20%,耐压刚好是 20 V 的合格二极管,那这个二极管很可能会被击穿。
- 因为以上一连串的如果,所以二极管被击穿的概率也不会非常高。不过这也是一个不稳定的因素。设计的时候,绝不能允许这种量级的损坏概率出现。

根据我的看法,绘制了图 7.5,大致描述这个问题出现的概率。Y 轴的概率是我凭借经验和感觉估计的,因为没有时间和精力做那么多的抽样实验。不过我想这样应该也能说明问题。

- 首先,如果肖特基二极管的耐压是 10 V 的话(好像 10 V 的肖特基二极管非常少)。如果有 1 000 个这样的器件安装在图 7.3 中,那估计至少有

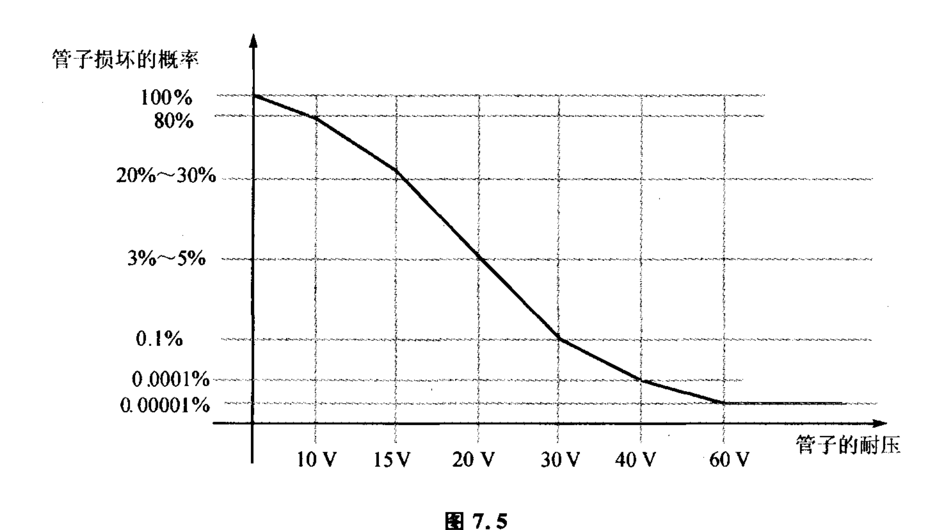


图 7.5

800～900 个会牺牲。因为这个时候，它的耐压明显低过 A 点的电压，损坏几乎是必然的。

- 当然，我也可以肯定的说：如果有 1 000 个这样的器件安装在图 7.3 中，也一定会有一些器件可以生存下来。因为在任何一个种群中，总会有一些很出众的个体，它们由于本身某些方面和大家有一些不同。所以它们可以耐受这样的恶劣环境。这也就是在古代，为什么无论是多么厉害的天花和鼠疫，只要有一定的人口基数，那每次瘟疫就不能消灭所有人口的道理。不过话说回来，这样的事情好是好，但是概率太低了，不是设计所想要的。
- 随着肖特基二极管耐压的逐步提高，其损坏的概率也会大幅度降低，到 40 V 的时候，估计就只有百万分之一的损坏概率。
- 继续提高肖特基二极管的耐压，其损坏概率仍然会下降，但是下降的幅度非常有限。它会逐渐趋向于一个大于 0 的很小的常数。这个常数就是肖特基二极管的自然损坏概率。这个常数也不会是 0，因为任何器件在额定条件下使用，也仍然有损坏的可能，虽然这种情况出现的概率极小，但仍然是有可能的。正如同某人坐在家里看电视，天上会突然掉下个失控的飞机一样。
- 所以在图 7.3 中，将肖特基二极管的耐压选定在 40 V 比较合理。不仅考虑了机器的安全性，也同时考虑了机器的成本。(因为耐压更高的器件要更贵一些)

以前一个同事设计的一个马达控制电路板，使用钽电容作为电源退偶。这个电路板使用13.5 V供电，电源进入电路板后第一件事就是退偶。而这个电路板的设计者使用了16 V的钽电容和104瓷介电容并联作为退偶。按理说这样的设计也没有什么大问题，但是这个电路板在生产200～300片后，有5%～6%的概率，老化后会出现16 V钽电容击穿的问题。虽然我们承认这个钽电容的耐压估计也有问题。但是考虑到钽电容击穿和电解电容击穿不同，钽电容击穿表现为短路，然后发热烧毁，而电解电容击穿是发热，然后是爆裂开路。二者造成的后果不同。按照前面所述的“权重”理论，短路故障的权重当然要高过开路故障的权重。所以将16 V钽电容的耐压提高一档到25 V。这样更改后，问题就再也没有出现。

俗话说：“常在河边走，哪有不湿鞋的？”这个俗语其实也是概率论的一个最简单的生活写照。想象一下，你站在沙滩上的时候，站在沙滩的浪花和沙滩交界线的最高处一个小时，被浪花湿润的概率就几乎是100%，后退1 m，概率立即会下降到50%（因为在随机抽样的任一时刻，涨潮和退潮的概率各为50%）。后退10 m，被浪花湿润的概率就应该不会高过1%了。再后退500 m，那估计只有前几年印度洋海啸那样级别的海浪才有可能打湿脚板了。不过这样的概率太小了，绝大部分的人，一辈子都碰不上一次。后退500m，即使被这样的可以载入史册的巨浪打湿脚板的话，也不至于会被海浪卷到大海里面去。所以相对而言，如果要绝对安全的话，就是这种状态。因为数学意义上的绝对安全是不存在的，有这样小的概率，基本就是工程技术上所谓的“绝对安全”。



这就是设计中的概率论：

- 你不是神仙，所以不可能确切知道危险来临的时候和危险的级别，但是可以根据经验和一些前兆事件，大致估算危险来临的时候和危险的级别。
- 计算好概率，远离危险！
- 计算好概率，用最小的资源消耗达到目的！

8

线缆的学问

电线电缆是硬件电路和电子系统中再平常不过的东西，很多人甚至不习惯把它称为元件，很多人只是将电线电缆片面的理解成将别的电路元件连接起来的东西，就像安装电灯一样，连接通了，电灯一定会亮，可是电线电缆真的这么简单吗？

如果你做硬件设计久了，才能真正理解，其实电线电缆其实也是很麻烦的东西，因为会有很多设计方面、使用方面的故障和毛病来自于这里。

8.1 为什么线缆是故障高发元件

8.1.1 折 弯

因为折弯而导致的电线电缆故障大部分来源于一种叫做“金属疲劳”的物理效应。那什么是“金属疲劳”呢？

人们所见到的金属，看起来熠光闪闪、铮铮筋骨，被广泛用来制作机器、兵刃、舰船和飞机等等。其实，金属也有它的短处。在各种外力的反复作用下，可以产生疲劳状态，而且，一旦产生疲劳就会因不能得到恢复而造成十分严重的后果。实践证明，金属疲劳已经是十分普遍的现象。据 150 多年来的统计，金属部件中有 80%以上的损坏是由于疲劳而引起的。在人们的日常生活中，也同样会发生由金属疲劳带来的危害。一辆正在马路上骑行的自行车突然前叉折断，造成车翻人伤的后果。炒菜时铝铲折断、挖地时铁锹断裂、刨地时铁镐从中一分为二等现象更是屡见不鲜。

所谓金属疲劳，就是一个机械零部件或结构件，如飞机、汽车、拖拉机以至桥梁的某一金属构件，只要反复承受一定变动的力，就会给金属表面特别是尖角、孔洞等应力集中点首先产生一个微小裂纹；由于承受力的继续反复作用，微小的

裂纹逐渐扩展，由小变大，到一定程度就突然断裂。通俗地说，人们大概都有这样的经验：当一下子折不断一根铁丝时，只要反复折几次，铁丝就断了，这就是铁丝已疲劳的缘故。

为什么金属疲劳时会产生破坏作用呢？这是因为金属内部结构并不均匀，从而造成应力传递的不平衡，有的地方会成为应力集中区。与此同时，金属内部的缺陷处还存在许多微小的裂纹。在力的持续作用下，裂纹会越来越大，材料中能够传递应力的部分越来越少，直至剩余部分不能继续传递负载时，金属构件就会全部毁坏、断裂和折断。

1998 年 6 月 3 日，德国一列高速列车在行驶中突然出轨，造成 100 多人遇难身亡的严重后果。事后经过调查，人们发现，造成事故的原因竟然是因为一节车厢的车轮内部疲劳断裂而引起的。从而导致了这场近 50 年来德国最惨重的铁路事故。

电线和电缆的金属疲劳产生的故障是相同的：电线的金属芯折断。产生这种故障通常有 2 个原因：过度折弯和反复折弯。

先说说过度折弯，电线的金属芯可以看作是一个金属条，通过施加外力可以改变它的形状，也可以通过外力将变形后的金属条再回到原先的形状，这样的反复，就是一个反复折弯的过程。折弯后，在折弯处都会形成一个或大或小的圆弧，这个圆弧的半径就是折弯半径。折弯的半径越小，电缆越容易折断，如图 8.1 所示。

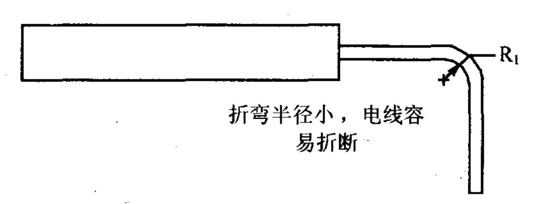


图 8.1

应该在塑料皮处折弯(见图 8.2)，而不要直接在金属芯处折弯，由于塑料皮的存在，包覆了电线的金属芯，可以防止电线金属芯的折弯半径过小。而且由于塑料皮的阻尼作用，可以消耗掉一部分折弯时施加在金属芯上的能量。所以总体上说，有包覆塑料皮的电线和折弯半径大的电缆可靠性要更好一些。

像图 8.3 那样的多股铜丝缠绕的软线，其抗折弯能力也大大好于同等截面积的单根芯线的电线。多股铜丝缠绕的软线其每根铜丝的直径都小得多，所以对同样的折弯半径来说，折弯半径和每个最小芯线的直径的比例就大得多。它

就比较抗折弯。由此可以发现，凡是电源线、USB电缆、AV信号连接电缆和耳机等需要经常插拔和经常卷绕的地方，使用的都是多股铜丝缠绕的软线，从来没有使用硬线的。而家里装修用的电灯线和电话线，这些地方的电线，安装好后就不需要动了，所以都使用硬线，因为软线的制造成本高于硬线。

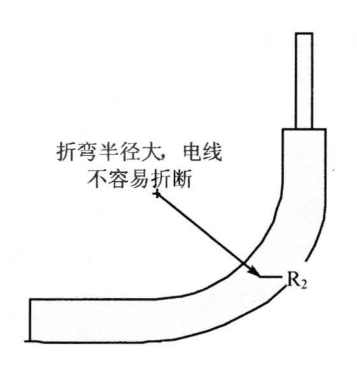


图 8.2

反复折弯也是产生金属疲劳的一个重要原因，即使是很大的折弯半径，但是只要达到足够的频度，一样可以折断金属条。用老虎钳反复弯折铁丝从而折断之，就是一个例子。一次折弯，铁丝一般不会折断，但是反复多次的折弯，任何铁丝都扛不住。像一些电动机的连接电线（高频率振动），手持设备的供电电缆和耳机电缆等都是属于折弯频度高的应用场合，而这些地方的电缆恰恰也是故障高发的地方。

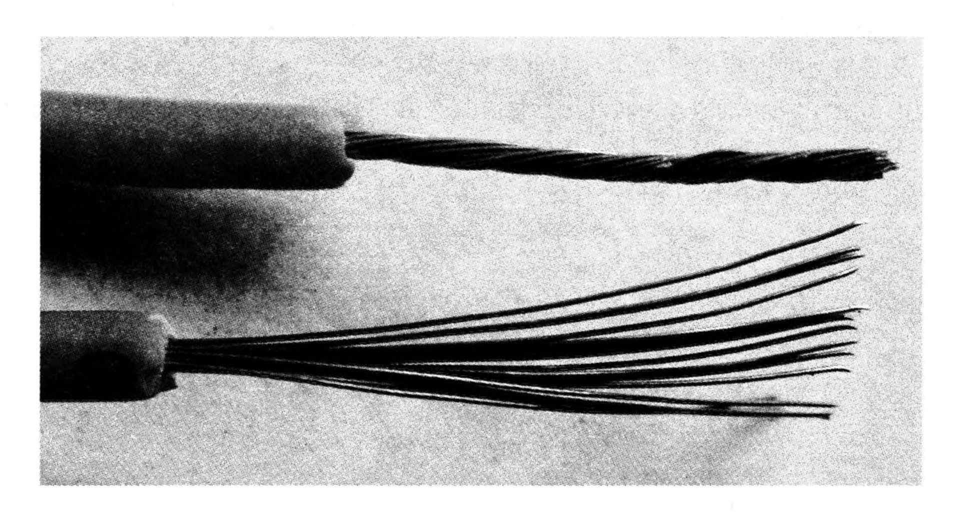


图 8.3

用线扣防止反复折弯，用线扣把电线固定在电路板的螺丝孔或机壳上，可以有效地防止电线被反复折弯，大幅度提高线路的可靠性。尤其是电缆折弯而失效的概率。在有电缆频繁折弯的场合，这个小小的线扣，成本不高，却可以减少很多麻烦。

在产品的设计过程中，经常要对电路板的裸板进行改动、测量及连接器插拔

等翻来覆去的操作，这样一些电缆尤其容易折断，用线扣把这些不需要折弯的电线和电路板上的螺丝孔等物体扣起来，可以大大节约设计时间。因为一个电缆断裂，你通常不会立即发现，而是首先体现出某种故障，等拿万用表、示波器找到这个故障点的时候，可能几个小时就过去了。这样的时间耗费，一点经济效益都没有。而在事前用一个价值 1～2 分钱的尼龙线扣扣好，就可以将发生这种倒霉事的概率降到最低。原先我没有这样的习惯之前，在日常设计工作中，每个月都要碰到 3～4 次这样电缆折断的事情，平均一个月要拿 3～4 个小时处理。采取线扣卡紧的措施后，这样无谓的时间耗费基本就没有了。如果把 3～4 个小时折合成人民币计算，这样一个小小的投入，对我这样职业做研发的人而言，至少价值 100 元/月。

图 8.4 和图 8.5 是一个电路板的实例，这个电路板使用以太网电缆做引出线，这里没有使用以太网连接器和水晶头，而是将以太网电缆的每个芯线用螺丝连接在脚距 5 mm 的大连接器上。由于以太网电缆的每个线芯是单根的铜丝，直径 0.3～0.4 mm 左右，可以折弯几次，但是很不耐反复折弯，在实际的使用过程中，由于电路板经常动来动去，这种电缆的线芯比较容易折断。将以太网电缆折回后用尼龙线卡扣在电路板的螺丝孔上，由于尼龙线卡限制了电缆的小半径折弯，所以以太网电缆容易折断的麻烦就可以迎刃而解。

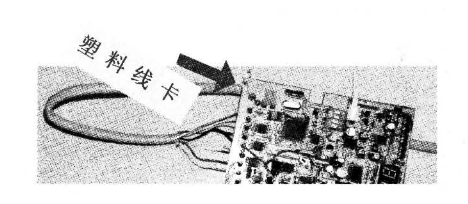


图 8.4

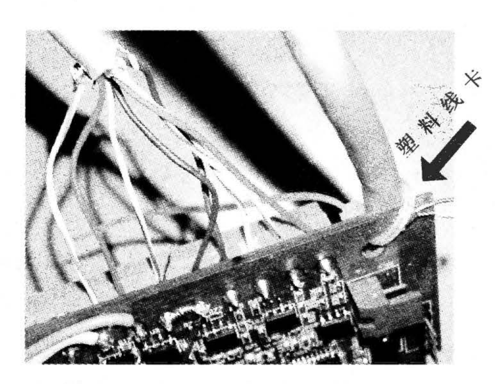
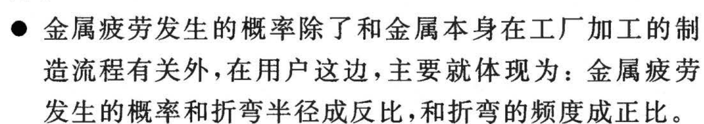


图 8.5





提示：

- 金属疲劳发生的概率除了和金属本身在工厂加工的制造流程有关外，在用户这边，主要就体现为：金属疲劳发生的概率和折弯半径成反比，和折弯的频度成正比。
- 记住：线扣不是可有可无的装配件，而是一种小投入大产出的预防性投入。使用尼龙线扣可以减少很多令人郁闷无比的低级错误。

8.1.2 电流流量超过载流量

普通的电缆都不是超导体,所以它们都有或大或小的电阻。一些应用中由于忽视了电缆的电阻,而恰好电路的电流又比较大,这样在线路上白白耗费掉了一些能量。更有甚者,电流的流量严重超过电线的最大标称容量,导致电线烧毁。不过好在电线一般对高温的耐受力还比较好,而且大电流的场合也不多,所以这一类的电缆故障总体较少。

8.1.3 特性阻抗导致信号畸变

这个问题在低频率的信号传输电缆上,不会体现出来,信号起码也要 500 Hz 以上才容易体现出这个问题。如果设计不良的话,那频率越高,这个问题会越严重。

特性阻抗是指当电缆无限长时该电缆所具有的阻抗。阻抗是阻止交流电流通的一种电阻。一条电缆的特性阻抗是由电缆的电导率、电容以及阻值组合后的综合特性。这些参数是由诸如导体尺寸、导体间的距离以及电缆绝缘材料特性等物理参数决定的。

很多电路,近距离传送信号一点问题都没有,一连上长电缆,马上问题一大堆,这些问题大部分是特性阻抗导致信号畸变和电缆间信号串扰引起的。比如,要传送一个 1 V/1 MHz 的方波信号到电路板的另外一端,是相当容易的。但是要经过 1～2 m 的电缆,信号就会开始振铃,(在本书的【注意你的高速数字信号】章节有专门对这个问题进行描述)而要把这个信号传递到 1 km 外去,那麻烦就大多了。1 V/1 MHz 的信号到 1 km 之外后,能够有 100 mV 就谢天谢地了。而且会由于电缆的分布电容和分布电感,将方波的上升沿和下降沿全部吃掉,只能得到一个接近于正弦波的波形。这还不算最要命的,最要命的可能是在 1 km 外,会窜入许多杂波,而这些杂波都是电缆作为一个天线,把天空中的垃圾信号给接收下来了。而这些都是只有用电缆传输信号才有可能发生的问题,短距离传输根本不会发生。

图 8.6 是一个 2 芯电缆的等效图,正是由于图中的等效电阻、等效电感和等效线间电容才造成了上述的问题。



图 8.6

所以信号线越长，越要考虑电缆可能造成的不良影响。

8.1.4 电缆间信号串扰

大家都知道，电话线都有 2 根芯线。可是有一个问题不知道大家想过没有，从电路的角度上讲，传送一个音频信号只要一根铜线就可以了，剩下的一根接地球的大地，一样可以形成环路，为什么都要用 2 根呢？这个问题是我刚刚学电子的时候反复思考的问题，曾经问过很多工程师，都没有得到满意的回答。最有道理的一个回答颇具哲学色彩："存在的就是合理的！"但是为什么要这样存在呢？

好在我在电信部门工作了很多年，这个问题以前不能回答，现在能回答得比较专业。为了说明这个问题，我把电话线的传输部分简化成如图 8.7 所示，这个图中，不包括交换部分，图中的"语音信号"就是电话局交换机出来的信号，但是电话局和您家里的电话可能还有 1～5 km 的距离，中间就只能用电话电缆来连接了。

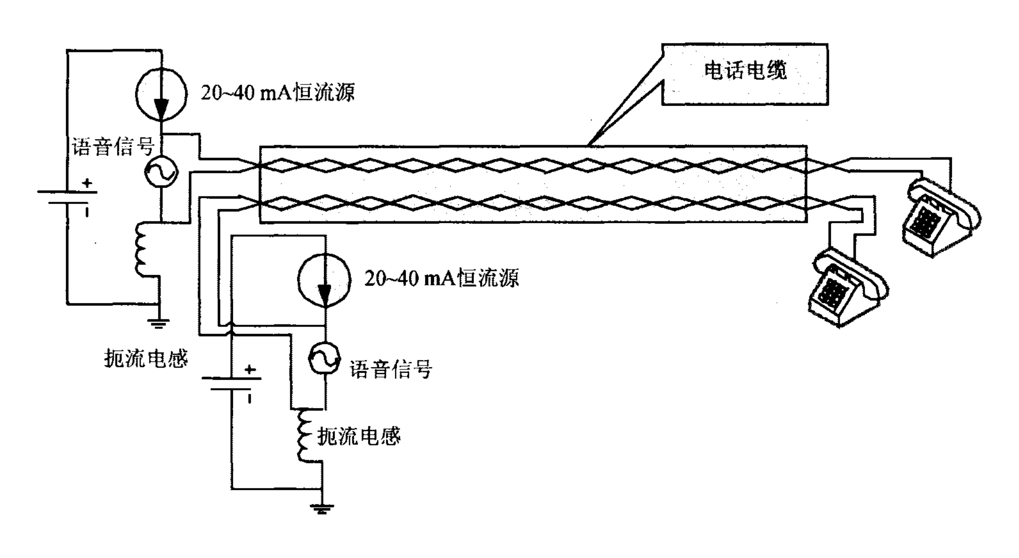


图 8.7

按照图 8.8 的连接应该最简单，最节约线材，两侧的电路也简单。但是事实上没有任何一个电话交换机使用这种形式的输出。因为像图 8.8 这样的电路，传递信号 50～100 m 不会有问题，但是长度如果达到 500 m～5 km，麻烦就大了，因为存在 2 个原因，这样的连接方式会导致在长距离电缆下通话质量变得不可接受：

- 一个电话电缆里面一般至少也有 16 对电话线，各个线芯之间都是互相缠绕或者平行的。长距离的缠绕或平行，会在线芯之间形成分布电容，距离越长电容越大，也就越容易将交流音频信号耦合到其他的电话线芯上去，造成串音。

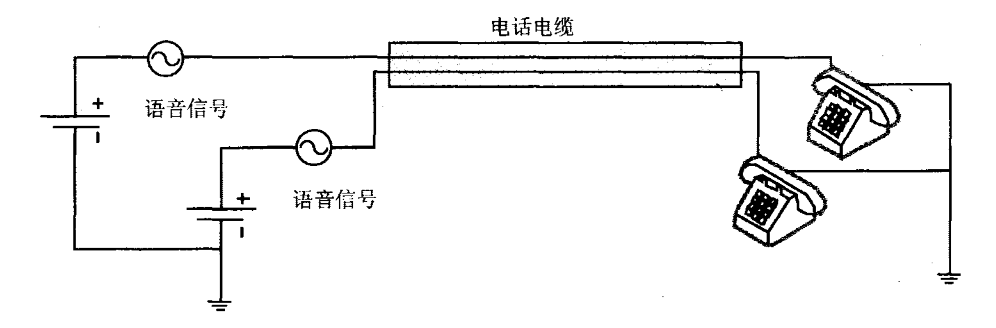


图 8.8

- 图 8.8 那样的连接为了节约一根电话线而共大地，但事实上，大地本身也有电阻，也可能产生干扰，架空的电话线构成了另外一半的回路，这一半很容易接收天空的长波、中波和短波信号，窜入话路造成干扰。而任何电子线路中都有非线性元件，比如 IC、三极管、二极管和生锈的连接器等，都可能对这样的广播信号检波，还原出广播信号出来，这样，在电话机中不仅能听见交换机送来的语音，还能免费听见天空的广播。

所以，现在的电话机和交换局之间的连接线都采用图 8.7 的形式，在交换局一侧，语音信号不直接接电源的正极或者负极，而是分别经过一个恒流源和一个电感才连接到＋24 V 或者＋48 V 电源。这样做的好处是让电源中的直流经过电话机形成一个回路，而同时不会因为供电而衰减掉语音信号。之后就是通过电话电缆中的一对一对的双绞线来将语音送到用户家中。为什么要用一对一对的双绞线来传送呢？关于双绞线，请看后文中关于双绞线的描述。

8.2 电缆的特性阻抗

电缆的特性阻抗是指当电缆无限长时该电缆所具有的阻抗。阻抗是阻止交流电流通的一种电阻。一条电缆的特性阻抗是由电缆的电导率、电容以及阻值组合后的综合特性。这些参数是由诸如导体尺寸、导体间的距离以及电缆绝缘材料特性等物理参数决定的。特性阻抗是一个对于交流电而言的参数，所以用

万用表不可能直接测量出一个电缆的特性阻抗。

正常的物理运行依靠整个系统电缆与连接器具有恒定的特性阻抗。特性阻抗的突变叫做特性阻抗不连续或特性阻抗异常,会造成信号反射,从而引起网络电缆中的传输信号畸变并导致网络出错。特性阻抗通常可以由电缆的连接和端结而造成轻微的改变。电缆的硬转弯或纽结也会改变电缆的特性阻抗。在不连续较轻的情况下,由于反射的信号微弱而且又经过电缆的衰减,所以对网络来说仍然能运行。大的阻抗不连续将会干扰数据传输。这类的不连续是由不良的电气连接、不正确的电缆端结、不匹配的电缆和不匹配的连接器的使用以及电缆中双绞电缆对的绞结方式错误而造成的。

不过对于电源线、电话线和音频线等线材来说,由于频率很低,一般不用考虑这个因素。但是视频线和射频线就要注意了,由于信号频率足够高,这些信号对特性阻抗非常敏感,对信号反射也非常敏感。比如如果视频线超过 100 m,再不注意传输阻抗的话,很容易造成重影等问题。

8.3 电缆与电缆的连接

电线焊接在一起,需要包覆绝缘层。这个道理大家都知道。但是我在长期的设计工作中,就发现不少电工和一些工程师(尤其是新手),随意处理电线接头,造成电线的连接不可靠。

以下是几种常见的处理电缆接头的方式:

- 焊接的时候,同样的电缆,不同颜色的芯线随意焊接在一起,虽不能说这个电缆不能用,但是这样的电缆在使用的时候,麻烦多多,因为焊接后的电缆两头的颜色都不一样,很容易因为看错线色而造成人为差错。(这样的事情,学校刚出来的年轻学生经常会干,他们通常会简单地认为:电线嘛,只要接通了就好了。)
- 电线焊接好了之后,不对连接的线头做绝缘处理(如图 8.9 所示),就拿这样的电缆做实验和其他用途。这样使用虽然绝大多数都是临时性的,但是裸露的焊点在做实验的时候很容易碰到电路板上,运气不好的话,轻则电路板罢工,重则烧坏电路板上的芯片。
- 焊接好线头后,如果用电工胶布包覆了绝缘层,可靠性已经好了很多。但是如果能够用热缩管来包覆绝缘层(如图 8.10 所示),可靠性就会更好。因为热缩管收缩之后,可以非常紧密地贴合其中的电线。

- 如果这样把电线折回一下，或者和一根没有断点的完整的电缆用线卡扣紧在一起(如图 8.11 所示)。虽然难看一些，但是抗拉扯的强度会有大幅度的提高，不至于因为强力的拉扯而将电线从焊点处再次断开。

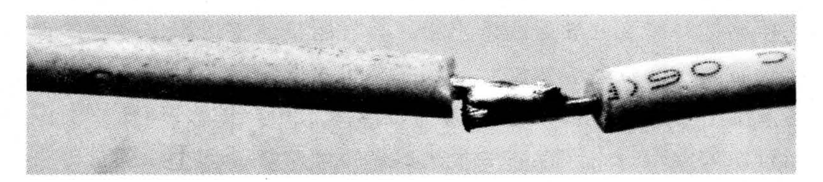


图 8.9

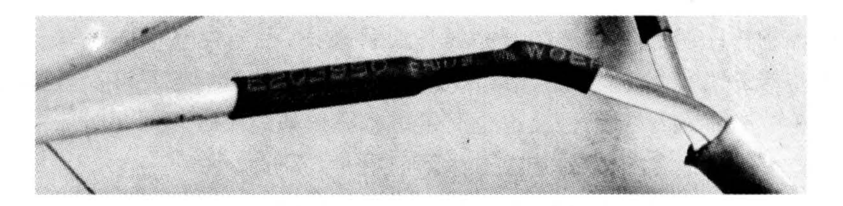


图 8.10

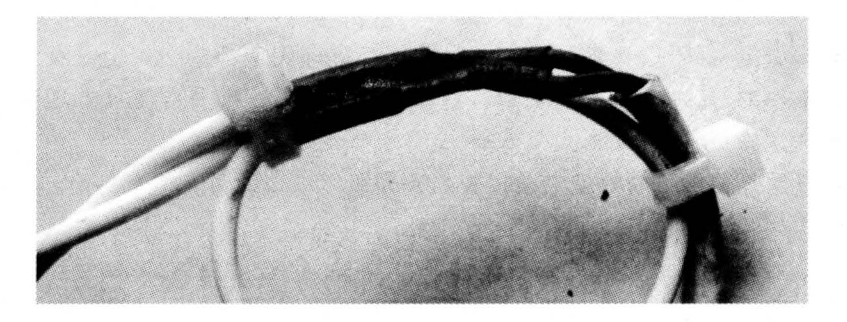


图 8.11

8.4 整机的线缆走线

- 电缆的颜色要符合业界的成文和不成文的规矩。比如电源正极一般用红色，电源负极一般用黑色等。如果乱配线缆的颜色，会给机器的维护和使用造成极大的麻烦。甚至有可能造成一些人为的故障。
- 电缆的各个芯线如果有顺序编号，一般第一号芯线要有明显标识。比如 IDE 硬盘的扁平电缆，第一 pin 就会有明显标识。

- 电缆要按规定的极性和标识连接，最经常犯的错误就是将电灯或一般电器插头、插座里面的火线和零线随意乱接，这样通常情况下不会有问题，但是比如一些日光灯关闭后，在非常暗的环境中，还会有一点微弱的光线，这种情况通常就是火线和零线乱接，导致开关虽是断开了零线，而火线依然连接着所致。这种情况虽然不影响使用，可是要有人去换灯管就会有发生危险的可能。因为这个时候即使电灯关闭了，整个日光灯和火线也是一直连通的。还有一种常见的情况就是在 AV 立体声设备中，L－R声道的电缆不按照规定的颜色和标示连接，导则声音左右颠倒，结果明明飞机从电视屏幕的左边飞往右边，但是还原出来的声音却是感觉飞机从右边飞向左边。
- 同样的信号回路，千万不要混用特性阻抗不同的电缆（使用特殊的阻抗匹配电路除外）。
- 在对同轴电缆和双绞线差分传输时，在电缆的终端节点一般要使用与电缆特性阻抗相同的端结器或者阻抗匹配电阻。端结电阻通过吸收信号能量以防止信号的反射。在视频电路中，这一个电阻就能有效防止重影。
- 在打开双绞电缆对安装连接器时或在配线架做连接时，非双绞的部分越短越好。
- 不要将电缆转小半径硬弯或打结。电缆的转弯半径应大一些。
- 在安装过程中要小心对待电缆。不要践踏电缆或过紧地捆绑电缆线。
- 强干扰线和信号线要隔开一定的距离，强干扰线包括电源线、CPU 的 IO 线、CPU 的总线、电动机和电铃的引线等，这些信号线由于可能包含频谱很宽的噪声信号，很容易辐射到附近的电缆。还有一些幅度很大的信号，比如 HI－FI 系统从音频功率放大器出来的音箱线，由于其电流和电压巨大，即使有很少的百分比辐射到小信号的电缆上，也会造成不小的干扰。实际经验表明，这两类的线材最好互相垂直走线，其影响就能做到最小。实在要平行走线，也要尽量拉开距离。
- 如果机器里面有很多平行的电缆，而它们在一起走，又不会造成干扰的话，可以用尼龙线卡把它们扎起来。这样，一根一根的电缆不仅不容易折断，也美观了不少。

8.5 双绞线

在上述电话的例子中，可能有人会说，用一对一对的双绞线来传送信号，是因为比较好管理，线序不会乱，颜色也好认。这样的回答虽正确，但却不全面。使用双绞线更大的理由是抗干扰。不仅可以防止别人干扰，也可以防止自己干扰别人。

双绞线是由一对带有绝缘层的铜线，以螺旋的方式缠绕在一起所构成的，通常的双绞线电缆是由一对或多对这样的双绞线对组成的，如图 8.12 所示。

绝缘材料使两根线中的金属导体不会因为互碰而导致电路短路。双绞线通常用于传输平衡信号。也就是说，在两条导线同时传输信号，但它们分别携带信号的相位相差 180 度。外界的电磁干扰给两条导线带来的影响将相互抵消，从而使信号不至于迅速衰退。螺旋状的结构也有助于抵消图 8.6 所示的不利因素，而如果是两根平行的导线就会形成一幅天线，不存在这种抵消效应，只有把它绞合起来才能发挥足够的抗干扰特性。

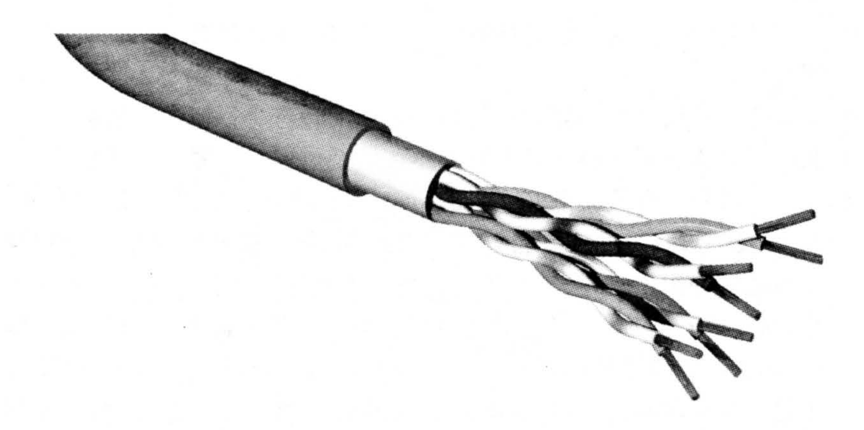


图 8.12 最常见的双绞线——以太网线

多对双绞线通常被捆扎起来，并外敷保护层。这样，成捆的电缆就可以被掩埋起来。与其他传输介质相比，双绞线在传输距离、信道宽度和数据传输速度等方面均受到一定限制，但价格较为低廉。很长一段时间以来，双绞线一直被广泛用于电话通信以及局域网建设中，是综合布线工程中最常用的一种传输介质。

虽然双绞线主要用来传输模拟声音信息，但同样适用于数字信号的传输，特

别适用于较短距离的信息传输。在传输期间，双绞线比较容易克服信号的衰减和波形的畸变。现在的以太网电缆就是采用双绞线技术的最好例子。

8.5.1 双绞线抗干扰的原理

作为信号传输的媒介，要求传输线不仅能有效地传输信号，同时具有很好的抑制干扰的能力。在双绞线中，干扰主要来自以下两方面：第一，外部干扰；第二，同一电缆内部各线对之间的相互串扰。下面，对双绞线消除干扰的原理作分析。

干扰信号对未扭绞的双线回路的干扰，见图 8.13。U_e 为干扰信号源，干扰电流 I_e 在双线回路的两条导线 L_1、L_2 上产生的干扰电流分别是 I_1 和 I_2。由于 L_1 距离干扰源较近，因此，$I_1>I_2$，$I_3=I_1-I_2\neq0$，有干扰电流存在。

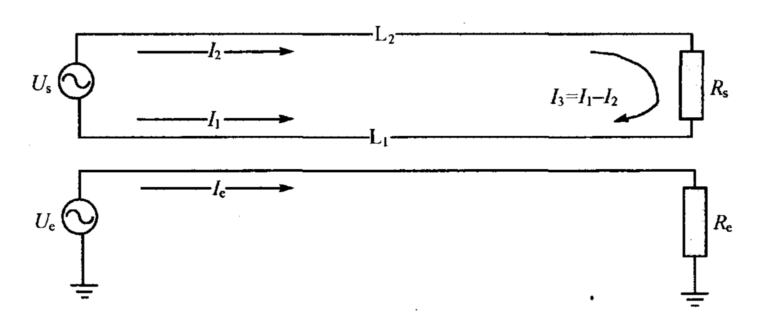


图 8.13

干扰信号对扭绞的双线回路的干扰，见图 8.14。与图 8.13 不同的是，双线回路在中点位置进行了一次扭绞。在中点的两边，各自存在干扰电流 I_1 和 I_2，$I_1=I_{11}-I_{21}$，$I_2=I_{22}-I_{12}$。由于两段线路的条件完全相同，所以 $I_1=I_2$。总干扰电流 $I_3=I_1-I_2=0$。通过分析，可以得出结论：只要合理地设置线路的扭绞，就能达到消除干扰的目的。

不知道您是否注意，像图 8.15 那样的以太网线缆，其内部的 4 个线对，其实有不同的扭距，肉眼不太容易看出来，但是把一段 5～6 m 的以太网线缆剥开，去数其中的 4 个线缆的扭转的次数，会发现 4 个线对各有不同，这样设计是为了防止各个线对之间的交叉干扰。同理，如果我们自己的线路中只要合理的设计扭距，就可以消除相互串扰。或者更简单一点，直接用以太网电缆来做信号传输电缆，也是不错的选择。以太网电缆不仅可以用于传递以太网信号，其实传递音频和视频也能做到非常好的效果。

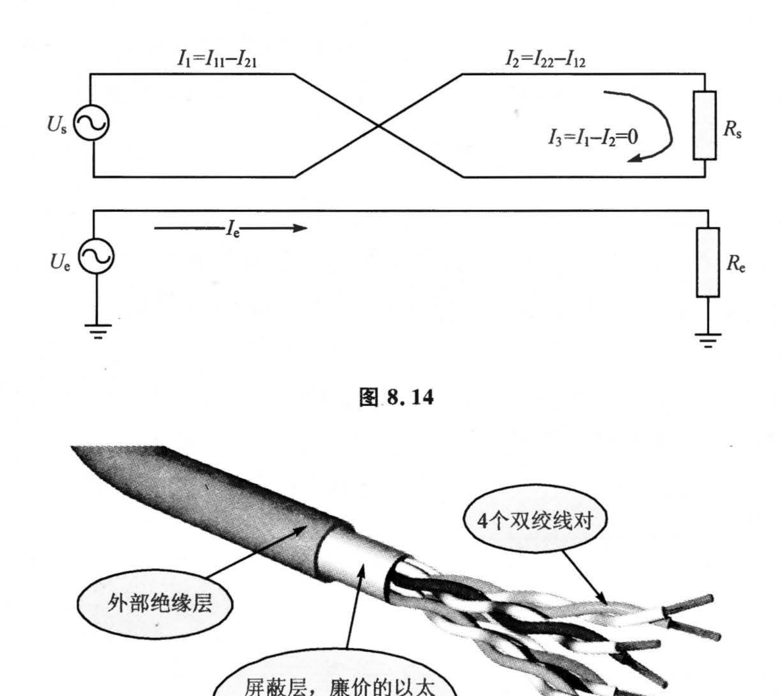


图 8.14

图 8.15

8.5.2 双绞线的特征

区分或评价各种类型双绞线的特征主要包括：导线直径、含铜量、导线单位长度绕数和屏蔽措施等，这些因素的综合作用决定了双绞线的传输速率和传输距离。

- 导线直径：即是铜导线的直径，一般直径越大，带宽就越高；
- 含铜量：直观的表现就是导线的柔软程度，越柔软的导线含铜量越高，传输能力越强；
- 导线单位长度绕数：表示导线螺旋缠绕的紧密程度，单位长度内的绕数越多，对干扰的抵消作用就越强；
- 屏蔽措施：屏蔽措施越好，抗干扰的能力就越强。根据双绞线缆是否带有金属封装的屏蔽层可以把双绞线分为非屏蔽双绞线(UTP)和屏蔽双绞线(STP)，如图 8.15 所示。理论上，屏蔽双绞线的传输性能更好，但

在实际使用中,屏蔽双绞线对于工程安装的要求较高,而且如果金属屏蔽层的接地不好,有些条件下其性能甚至还不如非屏蔽双绞线。因此,被广泛使用的实际上是非屏蔽双绞线。

8.5.3 双绞线的分类

双绞线传输模拟信号带宽可以达到几个乃至十几个 MHz,而传输数字信号的数据速率随距离而不同。EIA/TIA 为双绞线电缆定义了不同的规格型号,根据双绞线所支持的传输速率,主要可以分为以下几类:

- 一类线:由两对双绞线组成的非屏蔽双绞线。频谱范围窄,主要用于传输语音,而较少用于数据传输,最高只能支持 20 kbit/s 的数据速率。
- 二类线:由四对双绞线组成的非屏蔽双绞线。主要用于语音传输和最高可达 4 Mbit/s 的数据传输。
- 三类线:由四对双绞线组成的非屏蔽双绞线。主要用于语音传输和最高可达 10 Mbit/s 的数据传输,10 base - T 的以太网,即是采用三类线。
- 四类线:由四对双绞线组成的非屏蔽双绞线。用于语音传输和最高达 16 Mbit/s 的数据传输。
- 五类线:由四对双绞线组成的非屏蔽双绞线。用于语音传输和高于 100 Mbit/s 的数据传输,主要用于百兆以太网,如用在 100 base - T 的以太网中。
- 超五类线:由四对双绞线组成的非屏蔽双绞线。与五类线相比,超五类线所使用的铜导线质量更高、单位长度绕数也更多,因而衰减更小、信号串扰更小、具有更小的时延误差,在使用 4 对双绞线同时用于传输的情况下,可以用于 1000 base - T 的千兆以太网。

8.5.4 双绞线的优点和缺点

双绞线具有以下优点:

- 低成本,易于安装。相对于各种同轴电缆,双绞线比较容易制作,它的材料成本与安装成本也都比较低,这使得双绞线得到了广泛的应用。
- 应用广泛。目前在世界范围内已经安装了大量的双绞线,绝大多数以太网线和用户电话线都是双绞线。

同时双绞线还有很多缺点:

- 带宽有限。由于材料与本身结构的特点,双绞线的频带宽度有限。
- 信号传输距离短。双绞线的传输距离只能达到 1 000 m 左右,这对于很

多应用场合的布线存在着较大的限制，而且传输距离的增长还会伴随着传输性能的下降；如果要比这个距离更长，还要保持一定带宽的话，一般使用光纤。

- 抗干扰恶劣环境能力不强。双绞线对于外部干扰很敏感，特别是外来的强电磁干扰和雷击。而且湿气、腐蚀以及相邻的其他电缆这些环境因素都会对双绞线产生影响。在实际的布线中双绞线一般不应与电源线平行布置，否则距离长了就会引入干扰；而且对于需要埋入建筑物的双绞线，还应套入其他防腐防潮的管材中，以消除湿气的影响。

8.5.5 双绞线的应用

- ISDN：窄带 ISDN 中的基本速率接口(BRI)和基群速率接口(PRI)常使用双绞线作为传输介质。
- xDSL：基于数字用户线路技术(DSL)存在着多种接入网络的解决方案，如 ADSL、SDSL 和 VDSL 等，它们的共同特点是通过使用调制和编码技术在双绞线上实现了数字传输，达到了较高的接入速率。但这些 DSL 技术又在通信距离、是否对称传输、最高速率和使用双绞线对数等很多方面存在着不同。根据本地网络状况、带宽需求和用户使用习惯等不同，它们有着不同的应用场合。目前在我国，非对称数字用户线路(ADSL)技术被大规模的用于接入网络建设中。在我国的电话网络中，特别是公共电话网络用户线路的布线中还存在着大量的平行线，在电话通信中使用平行线代替双绞线的影响不大，但当利用这样的接入线路作 ADSL 接入时，就会产生较大的影响。ADSL 下行的最大速率可以达到 8 M bit/s，而采用平行线替代双绞线一般只能达数百 K bit/s 的下行速率。
- 以太网：目前十兆/百兆/千兆以太网的主要传输介质都是双绞线，这其中，十兆/百兆以太网使用 2 对双绞线，千兆以太网使用 4 对双绞线，一般的以太网线都包含 4 对双绞线。部分以太网线也采用平行线或同轴电缆作为传输介质。

8.6 差分电路

本来电缆是电缆、电路是电路，电路不应该放在本章陈述。但是差分电路和电缆有密切的关系，所以还是将它放在这里比较好。

按照线路的形式，传输线可分为非平衡式和平衡式两种。同轴电缆属于非平衡传输线，双绞线属于平衡传输线。视频设备，包括信号源及显示设备，其接口通常是非平衡式，因此，可直接与同轴电缆匹配连接。要用双绞线传输平衡信号，必须在发送端将信号提供设备的非平衡信号转换为平衡信号，以便驱动双绞线，在接收端再将平衡信号转换为非平衡信号，与信号使用设备连接。一个基本的双绞线视频传输系统如图 8.16 所示。图中的 A_1 是差分信号发送放大器，完成非平衡到平衡的转换，A_2 是差分信号接收放大器，完成平衡到非平衡的转换。

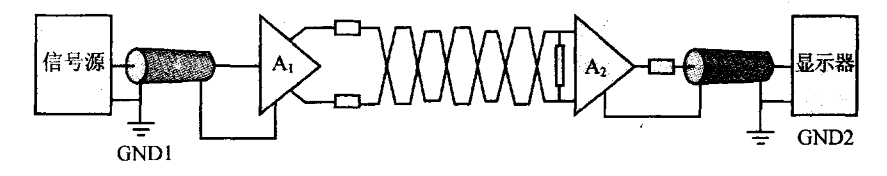


图 8.16

对于长距离电缆特性阻抗导致信号畸变的问题，可以使用电流环差分信号来传送数字信号。请参考“1 模拟，无处不在”中的相关内容。

8.7 同轴电缆

同轴电缆由中心的铜质或铝质的导体、中间的绝缘塑料层、金属屏蔽层以及主要起保护作用的外套层组成。这其中，同轴电缆的铜导体要比双绞线中的铜导体更粗，而接地的金属屏蔽层则可以有效的提高抗干扰性能。因此，同轴电缆具有比双绞线更高的传输带宽。同轴电缆的结构如图 8.17 所示；实物照片如图 8.18 所示。

同轴电缆中的屏蔽层既可以是铜质网状的、也可以是铝质薄膜状的，它的另外一个作用是防止寻找食物的饥饿啮齿类动物破坏裸线。绝缘塑料层和外套层均可以有不同的形状、结构和强度，这一般取决于电缆使用时的安装条件和使用环境等因素。例如，应用于室外环境的架空电缆由于会工作于强风以及雨雪等恶劣环境，因此需要强度较高的外套层。

同轴电缆的传输特性优于双绞线，这主要是缘于同轴电缆使用更粗的铜导体和更好的屏蔽层。更粗的铜导体可以提供更宽的频谱，一般可达数百 MHz。另外信号传输时的衰减更小，也可以提供更长的传输距离。普通的非屏蔽双绞线是没有接地屏蔽的，因此同轴电缆的误码特性大大优于双绞线，可以达到

10^{-9}。同轴电缆的这种结构，使它具有高带宽和极好的噪声抑制特性。实际应用中，同轴电缆的可用带宽取决于电缆长度。1 km 的电缆最高可以达到 1～2 Gbit/s 的数据传输速率。也可以使用更长的电缆，但是传输率就要降低或需要使用信号放大器。

常见的同轴电缆有两种：一种是 50 Ω 阻抗的同轴电缆，用于数字传输，由于多用于基带传输，也叫基带同轴电缆；另一种是 75 Ω 阻抗的同轴电缆，用于模拟传输，也被称为宽带同轴电缆。宽带同轴电缆在使用中其带宽可以被划分为几个范围。通常每一个频率范围都携带着各自的编码信息，这样就可以在一根电缆上同时复用地传输多个数据流。

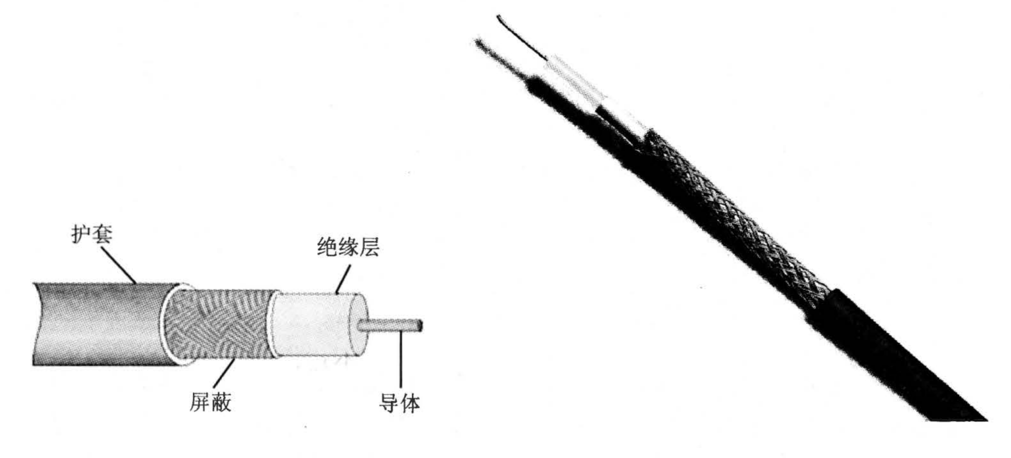


图 8.17　同轴电缆的结构　　**图 8.18**

同轴电缆的常见规格如表 8.1 所示。

表 8.1

规　格	类　型	阻　抗	描　述
RG－58U	细缆	50 Ω	固体实心铜导线
RG－58A/U	细缆	50 Ω	绞合线
RG－58C/U	细缆	50 Ω	RG－58A/U 的军用版本
RG－59	CATV	75 Ω	宽带同轴电缆，用于有线电视中
RG－8	粗缆	50 Ω	固体实心线，直径约为 0.4 英寸
RG－11	粗缆	50 Ω	标准实心线，直径约为 0.4 英寸



8.7.1 同轴电缆的特点

- 可用频带宽：同轴电缆可供传输的频谱宽度最高可达到 GHz，比双绞线更适于提供视频或是宽带接入业务，也可以采用调制和复用技术来支持多信道传输。
- 抗干扰能力强，误码率低，但这会受到屏蔽层接地质量的影响。
- 性价比高：虽然同轴电缆的成本要高于双绞线，但是它也有着明显优于双绞线的传输性能，而且绝对成本并不很高，因此其性价比还是比较合适的。
- 安装较复杂：双绞线和同轴电缆一样，线缆都是制作好的，使用时需要截取相应的长度并与相应的连接件相连。在这一环节中，由于同轴电缆的铜导体较粗，因此一般需要通过焊接与连接件相连。其安装比双绞线更为复杂。



8.7.2 同轴电缆的应用

同轴电缆以其良好的性能在很多方面得到了应用：

- 早期的局域网：早期的以太网大多采用同轴电缆作为传输介质，当用于十兆以太网时，传输距离可以到 1 000 m。很多生产年份较早的网卡均同时提供连接同轴电缆和双绞线的两种接口，不过由于其成本高过 5 类线，所以这几年在局域网中的应用少多了。
- 局间中继线路：同轴电缆也被广泛地用于电话通信网中局端设备之间的连接，特别是作为 PCM / E1 链路的传输介质。
- 有线电视(CATV)系统的信号线：直接与用户电视机相连的电视电缆多采用同轴电缆。这一电缆一般既可以用于模拟传输，也可以用于数字传输。在传输电视信号时一般是利用调制和频分复用技术将声音和视频信号在不同的信道上分别传送。这是同轴电缆在民用中用的最多的地方。
- 射频信号线：同轴电缆也经常在通信设备中被用作射频信号线，例如基站设备中功率放大器与天线之间的连接线。

9

尽量为后续的工作多遗留一些信息

我们有一个很漂亮的模具，制作一个 DVD 播放机，在研发过程中调试好了电路，做了几个样机，一切 ok。之后书面记录了第一次做 PCB 的所有软件和硬件的问题(bug)，提交相关工程师一一改进之。但是等第二轮的 PCB 回来，却发现，虽然记录下来的那些内容都改掉了。但是原来和模具没有偏差的 PLAY 按键，却鬼使神差的在 X 轴上偏移了大约 2 mm，在 Y 轴上也偏移了将近 1 mm。虽然画板的工程师也觉得很难为情，但他说确实不知道按键怎么就会从正确的坐标移动到了错误的坐标上。从他的眼神可以看出，他确实没有撒谎。

那为什么会移动呢？总结可能有以下几个可能性：

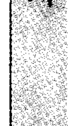
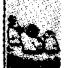
1. 这个按键没有在 PCB 软件中设置 lock 属性，所以可以被轻易移动，而 PCB 画图软件不会提出警告。这种移动可能是有意的移动，也可能是由于疏忽无意间的移动。
2. 移动后，没有有效的措施及时发现这个错误。

各位读者在继续看本章内容之前，需要理清楚这么几个原理：

- 每个人都会有疏忽的时候。
- 粗心的人和细心的人的差别不在于有没有错误的差别，而是发生错误概率的差别。
- 错误发生的概率除了人的因素以外，还和做事情的流程密切相关。有时候流程的因素要大过人的因素。

如果你承认以上 3 点，那就请继续往下看。

以上发生错误的第一个可能性比较容易解决，让画板的工程师知道，PCB 元件还有一个属性叫做 lock，设置了 lock 就不那么容易移动元件了。第二个可能性看似也容易解决，把责任工程师批判一下也行。但是这里面有这样一个问题，虽然这个问

题可以完全归咎于责任工程师的工作细致与否，但是通过制度完善，也可以解决。毕竟天下没有人是会故意犯错的，很多错误都是因为缺少某方面的知识、经验和信息而无意中犯下的。

所以，我们对 PCB 的绘制流程做了以下改进：

- 每个 PCB 文件上都要打开机械层，在里面标注上模具各个地方的限制高度，模具开孔的投影线。这样就将 3 维的模具，有关于电路板的信息都记录在了机械层上。
- 画板的时候，一定要打开这个机械层，这样元件移动的时候，可以清楚的看见是否有风险。
- 在机械层上做标注一定要详细。

在改进工作流程后，就极少发生类似的问题了。

这样一个流程，其实说明了一个问题，就是尽量为后续的工作多遗留一些信息。这个后续的工作可以是自己的后续工作，也可以是别人的后续工作。像上面的例子，因为有 3 维模具在 PCB 上的 2 维投影信息，所以后面的电路板，无论是他自己画还是换一个人画，都不容易出差错。

类似的，其实很多地方都是这样做事情的，比如：

- 写软件的时候要求尽可能详细的对自己的软件进行注释。
- 把软件模块提交给别人，要写一个模块的 API，告诉人家如何调用自己的代码，你写的越详细，后面的工程师的开发速度越快。API 做的最好的例子就是 Microsoft 的 MSDN，我看过的软件 API 里面最好的就是 MSDN，其次是 Borland 的 Help，但是 Borland 因为别的原因日渐衰落，想起它的漂亮的 help 文件，就会开始怀念 Borland。
- 医院里面看完病了之后，一般都要写病历。这就是为后续的治疗留下方便。可以想象一下，比如，你今天发现注射青霉素皮试过敏，医生偷懒没有把它记录在病历里面。明天上班再换一个医生，不知道你的情况，因为这是人命关天的事情，所以他也不会理会你说的“昨天做过皮试了，阴性！”。而只会按照通常的程序再给你做一次皮试，这个时候你会不会很郁闷？

总之：

- 于公来说，给后续者留下尽可能多的信息，会让后续工作者感到愉快，出问题也少。
- 于私来说，因为后续工作者出问题少，所以你的耳根子一定会清净许多。
- 即使这个后续工作者是你自己，也同样会降低你设计中的 bug 发生概率，也同样是造福你自己。
- 为后续的工作多遗留一些信息，本质上是牺牲眼前的一些小的利益，以降低将来可能发生问题的概率。
- 后续的工作者如果因为信息的缺失而造成的风险兑现，其损失经常会远远超过你将工作细化的时间。

10 搞定噪声

10.1 低阻抗！低阻抗！

经常在 EDN BBS 上逛荡，曾经回答了一个模拟电路的问题，是解决一个电路的自激振荡。比较有代表性，特写出解决问题的思路，和大家分享。

以下是该帖子的原文，为便于理解，对原文作了一些修改：

我用 OPA637A 作了一个有耗损的积分电路。（以前在这里问过关于积分电路的问题，再次谢谢当时回帖的各位）调整反馈电容与电阻的搭配本来一切都正常，可当我接入了一个 buffer LH0002Ch 或 CLM6321 作为同轴电缆驱动的时候电路就出现振荡。

后来我发现把反馈电容调小，震荡信号会变小，最终我用了 3 pF 的反馈电容总算稳定了下来。可这样就不得不调大与之并联的电阻。这也不是长远之计啊！

后来还发现当输入端接入一个电容性的平板天线后这种振荡的情况又会出现。震荡的频率为 20 MHz 到 30 MHz 的正弦波（频率不稳定），不知道是什么引起的。

电路图如图 10.1 所示。

我仔细看了这个电路，然后给出了如下回复。

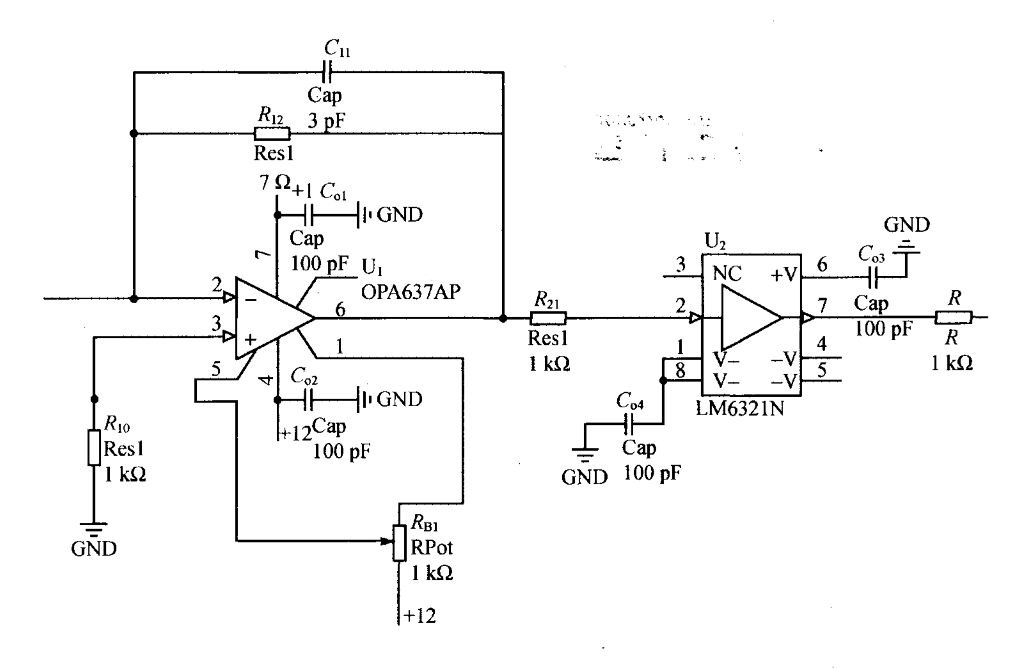


图 10.1

> 在 R_{21} 和 U_2 之间的那个网络上，对地并联一个电阻看看，电阻值分别选 50 Ω、75 Ω、150 Ω、510 Ω、1 kΩ、10 kΩ。用示波器看看效果。如果不允许直接接电阻(可能会影响电路的直流工作点)，那就在电阻上在串联一个小电容。
>
> 可能是信号的反射，如果是的话，那我提出的办法，估计就是一个解决的办法。

2 天后，对方回帖，说：按照我的方法，可行。电阻最后选定了 50 Ω。并表示感谢。这个时候，我也非常开心。更改后的电路如图 10.2 所示。

我在 BBS 中帮助网友解决这个问题，但是并没有在 BBS 中说明解决问题的思路，我思维过程是这样的：

- 首先第一个理由，电路进入了不稳定状态。因为振荡的频率和幅度都是不稳定、不可控的。
- OPA637A 这个运算放大器，看似被连接成为负反馈。但有可能某些特定频率的信号其实是正反馈。
- 楼主说：**“后来我发现把反馈电容调小，震荡信号会变小，最终我用了 3 pF的反馈电容总算稳定了下来”**。虽然这个也说明了更改负反馈电容可以解决一些问题，但是由于电路板分布参数的差异，所以不代表这个

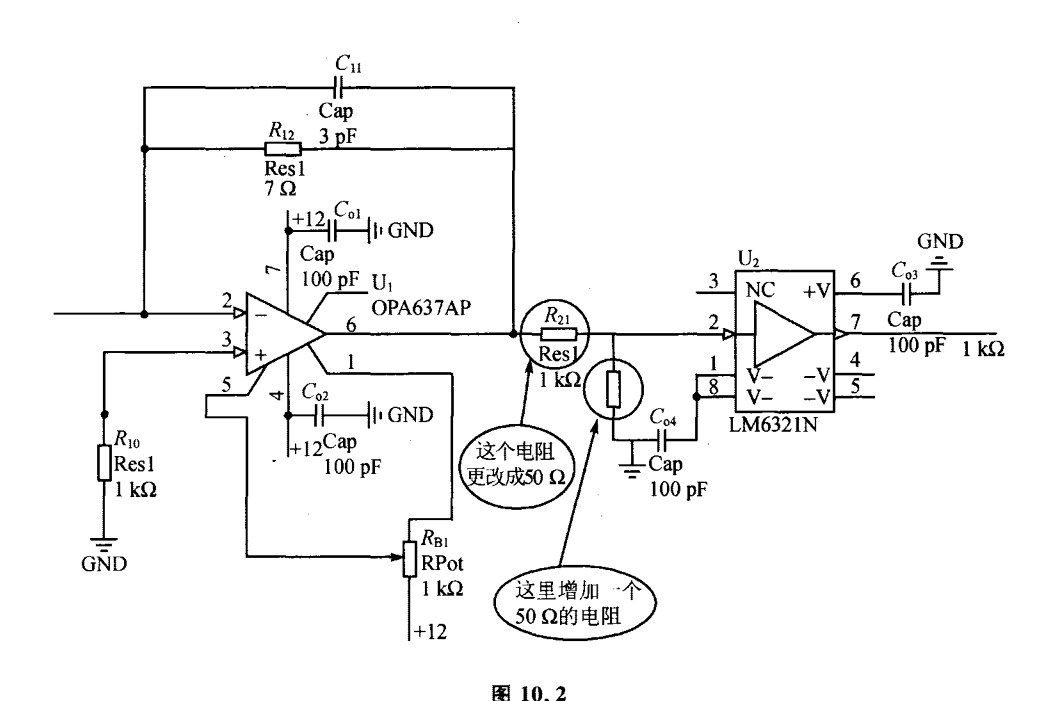


图 10.2

电路板生产 100 片、1000 片，都可以用 3 pF 的电容来解决问题。也不会因为电路板今天可以因为这个 3 pF 电容稳定下来后，明天还能继续稳定。

- 把负反馈电容减小（楼主还说“**可这样就不得不调大与之并联的电阻**”），其实是减小了某些高频信号的回授电平。当某些频率的型号在这个电路形成正反馈后，减小这个电容和增大这个电阻，都可以减小正反馈。导致寄生振荡暂时削除。
- 更改了 C_{11} 和 R_{12}，对于有用的信号，它的负反馈也少了，同样会导致不稳定。所以我说“不代表这个电路板生产 100 片、1000 片，都可以用 3 pF 的电容来解决问题”。
- 所以更改 C_{11} 和 R_{12}，是一种治标不治本的方法。

那，为什么我的方法可以解决问题呢？先看看图 10.3，这是运算放大器 OPA637A 简化后的等效图。不管你愿不愿意承认，以下几点都是客观事实：

- 任何信号它都有一定的信号源内阻，包括有用的信号和没用的噪声信号。
- 任何放大器都有输入端和输出端，输入端有输入阻抗，输出端也有输出

阻抗。

- 任何放大器都有本底噪声、任何元器件都有热噪声、电路板也会接收天空的一些电磁波感应出噪声来。这些都是图 10.3 中所表述的"噪声信号"。
- 由于信号频率、相位、在电路板上的布局、元器件额定参数和分布参数等原因,某些特定的信号在一个负反馈电路中,有可能会成为正反馈。

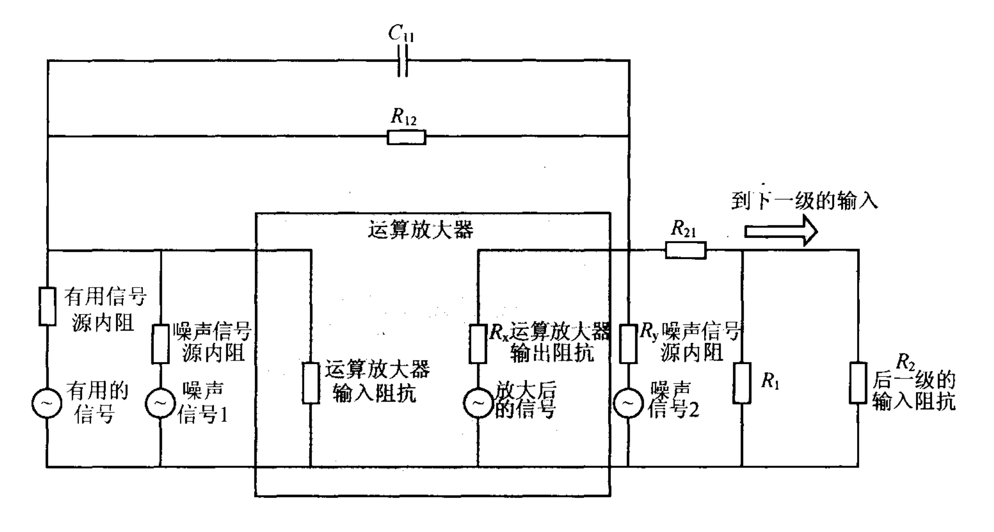


图 10.3

图 10.3 中的 R_1,就是新增加的 50 Ω 电阻。

- 在运算放大器的输出端,同样也有输出阻抗,在图 10.3 中用 R_x 表示。现在假设它的输出阻抗是 1 kΩ。
- 假设后级放大器的输入阻抗 R_2=10 kΩ,10 kΩ 的假定值已经很低了,实际上大部分放大器其输入阻抗都大过 100 kΩ。
- R_1 没有连接(就是楼主最早的那种状态)。
- 假设放大后的信号有效值是 1 V。
- R_y 是运算放大器输出端一侧的噪声信号的等效信号源阻抗,一般噪声信号源的内阻都很大。现在假定它等于 10 kΩ。这个信号可以是前面描述的"放大器本底噪声、元器件热噪声、电磁波感应噪声",也可以是由于信号反射引起的振铃噪声。
- 在以上条件下,根据电阻分压公式,可以粗略计算一下,接入了电阻 R_1 后,无论是反馈到前端的信号还是给后级放大器的信号,其噪声电平都

将被大大衰减，而有用信号虽然也同样会被衰减，但是由于其输出阻抗比较低，被衰减的幅度不大。

- 即使电路中还保留了某些特定频率的正反馈通道，但是只要把正向反馈的幅度控制在不足以引起链式反应的水平，电路就永远自激不起来。
- 当一个电路中没用的噪声信号被控制住了以后，电路自然就趋于稳定。

类似的问题，还有一个案例，一个音频噪声的案例。先看图 10.4，这个电路是原先我公司一个同事设计的电路，工作得也还算可以。只有一个比较讨厌的毛病，就是在用手调节电位器（手指没有直接接触到电路，只是旋转电位器的塑料手柄）的时候，喇叭里面会有一定程度的“嗡嗡”交流声。量产的机器都有同样的毛病。

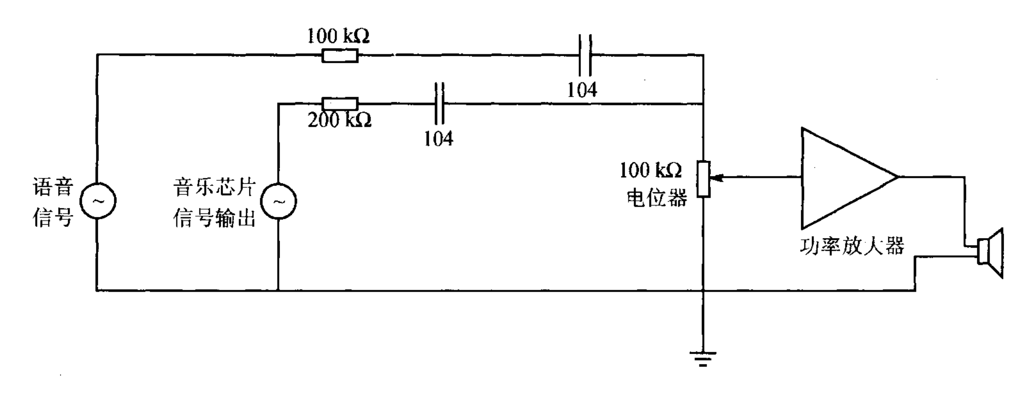


图 10.4

我看完这个电路后，基本明白他设计错在什么地方了。对于电位器来说，前面的 100 kΩ 电阻和 200 kΩ 电阻就相当于信号源内阻。而噪声源就是人的手（等效电路见图 10.5），人体自身是导电体，在空间电磁波和空间磁场的辐射下，会产生感应电压。所以也是一个比较强的噪声信号源，当它靠近电路后，通过分布电容和空间辐射来影响电路。只要能平衡好信号和噪声的比值，就很容易解决这个问题。

于是我开出以下处方，该处方不需要更改电路板，而且对于生产部门来说非常简单。开出处方的依据和前面那个案例类似：

- 将 100 kΩ 电阻改成 2 kΩ。
- 200 kΩ 电阻改成 4.7 kΩ。
- 104 更改成 10 μF。
- 100 kΩ 电位器改成 2 kΩ 电位器。

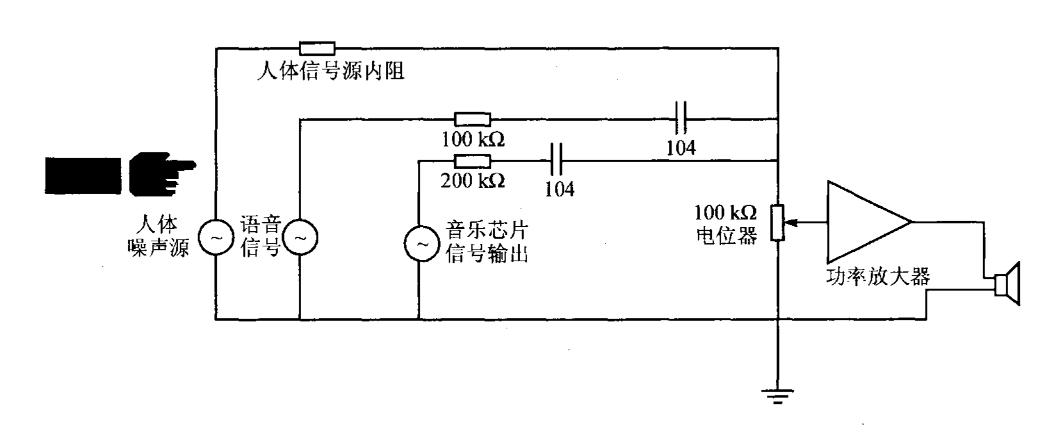
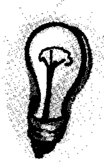


图 10.5

这样更改后，整个电路的阻抗都降低了，而人体噪声源的阻抗没有变化。所以，人体噪声被大幅度衰减。而有用信号的幅度却没有多大改变。于是就理所当然的解决了以上问题。



一句话总结本节的核心思想：

把电路设计成低阻抗，可以有效降低噪声对电路的影响！因为噪声源大多是高内阻的。

10.2 恐怖的火花干扰

多年之前，我的一个同事受命设计一个单片机控制的告警柜。这个设计任务只有一台订单，虽然没有量，好在这一台客户给的价格不菲，公司还是接下了这个订单。

这个系统可以输入 10 个模拟量和 20 多个数字量，主芯片是一个 AT89C55。这个系统的输出有：5 个告警灯，2 个电铃和 5 个继电器输出。

这个系统在设计软件的时候，那个年轻帅哥工程师嫌电铃太吵，将一个 510 Ω的电阻和一个 LED 串联起来代替 12 V 的电铃，这样，电铃在有输出的时候，就可以看见 LED 点亮。

软件编写和测试一路顺风。测试好了以后，就开始总装整机了。总装的时候，按照客户的订货合同，将连接器上焊接的 LED 拿掉，通过连接器连接了一个

电铃。

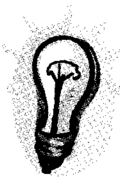
总调试开始后，调试人员总是报告："电铃一响，机器就再也没有反应了！"。把那个年轻的帅哥工程师抓到安装现场，他看了也犯晕。"我明明测试过的呀！"，他一直这样反复说明着。在现场忙碌了一阵后，实在没有什么效果。就把那块带着软件的电路板拆卸下来，回到实验室再测试一次。

在实验室里，他把电路板恢复到了原来的开发环境。软件跑起来后，运转一切正常，代表电铃 LED 依旧会亮起，之后可以切断代表电铃告警的 LED；可以响应下一次的告警。实在看不出软件有什么问题。但是这个电路板在安装现场，故障还是电铃一响，电路板就死机。

这样的情况，当时谁也说不上来什么原因，只好麻烦那个小帅哥用笔记本电脑把那个软件带到现场调试，当用仿真器运行他的软件后，进到告警程序之前，一切正常，只要软件一打开了那个电铃，单片机程序就会跑飞，而且每次跑飞的地址都不一样。

我们在折腾了将近 1 天，排除了无数种可能后，实在想不出来还有什么样的可能，就拔下电铃，把那个代表电铃的 LED 连接到电路中，发现居然一切正常。再换上那个电铃，故障依旧。反复试了几次，居然百发百中。发现这个规律后，大致就明白了：电铃能够响起来，在于其内部有机械/电磁震荡，由于其内部有电磁铁驱动接点，在响铃的时候该节点会高频率的反复开关，又由于这个开关驱动的是线圈这样的感性负载，开关附近会有火花，就是这些火花导致了死机。

为了证实我们的猜测，将电铃更换成了电阻或灯泡等负载，都不会导致死机。最后，我们用一个 12 V 的带筒形喇叭的高音量电子蜂鸣器代替电铃，固定好后。老化测试一周都没有再次死机。



为什么电子蜂鸣器可以在这个例子中替代电铃？

一般的带电磁铁的电铃，在工作的时候，它的金属触点会反复通断，通断的频率超过 10 Hz。而这个触点驱动的又是线圈那样的感性负载。在接通和断开的瞬间，电感线圈都会产生感生电动势，导致触点上有火花。电火花是一个很强的噪声源，有广泛的频谱分布和很强的能量。在这个系统中，单片机的电源是从

24 V降压下来的，而电铃也是直接从24 V上取能量的。火花产生的干扰，会通过电源以有线的方式串入单片机的电源，或者以无线的方式将噪声辐射出去，被PCB上的铜箔吸收后，产生干扰电压。最后导致单片机的复位和电源产生异常。

同样的道理，如果这个时候驱动的是一个碳刷电动机，也一样会有火花。可能也一样会导致死机。

电子蜂鸣器不存在这样的触点和电火花，所以产生的干扰也小得多。即使它是那种高分贝的蜂鸣器，也不至于向外辐射太多的噪声信号。所以它在这个系统中可以安全运行。

所以只要是不产生火花的东西，比如电阻、电子蜂鸣器、甚至是不产生火花的无刷电动机，在这里都可以作为安全的负载。

我还处理过一个更麻烦的案例。如图10.6所示，这是一个控制瓦斯点火的电子设备。开始还以为这个电路很简单，很快就把原理图画好，PCB做好了。和上面的案例一样，开始一切都好，可是等电子点火器一接上，麻烦就来了。只要电子点火器一开通电源，随着噼噼啪啪放电的火花出来，CPU经常也会死机。即使把点火器用电缆拉到15 m之外也无济于事。

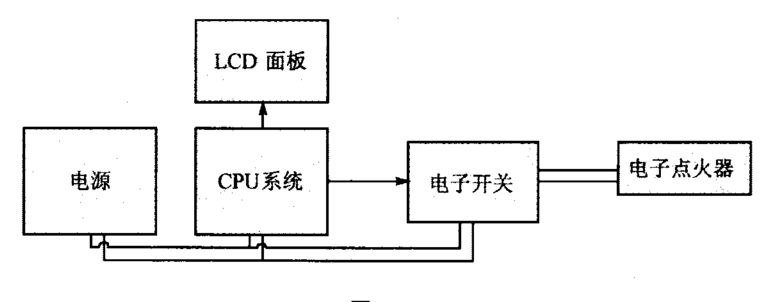


图 10.6

按照以上案例的思维，首先想更换一个不会有火花的点火器，可是这个问题提出来，我自己都觉得好笑："这个世界存在着没有火花的点火器吗?"。上网搜索了一大圈，也确实没有发现有这样的点火器。所以只能死了这条心。

在不能消灭火花的前提下，为了搞定这个问题，尝试了以下方法：

- 用马口铁将电子点火器几乎全部封闭起来，只留下2根电源线和2个点火触电。无效。
- 将以上电子点火器的屏蔽马口铁接电源地，无效。



- 将 CPU 系统和 LCD 面板等敏感部件用马口铁屏蔽，也无效。
- 再将电子开关也屏蔽起来，还是无效。
- 按照图 10.7 所示，将电子点火器的电源用 DC/DC 隔离电源，隔离开来。死机的概率大幅度下降，但还是会有死机的可能。可靠性还是一个问题。

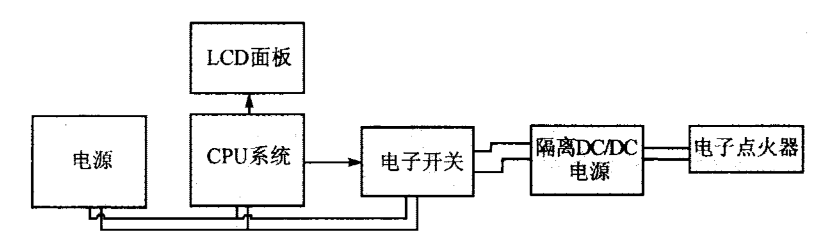


图 10.7

最后想了很多办法，按照图 10.9 所示，在电子点火器前面插入一个绕线磁环，这样处理后，死机的问题居然彻底根治了。

之所以火花干扰，如此难以搞定，是因为在火花中，包含广谱的无线电射频干扰信号，这些干扰信号不仅频域宽广，而且辐射强度非常大。最容易通过天空辐射和导线进行传导干扰。在这 2 种干扰中，天空辐射还比较容易搞定，只要电子点火器和主机拉开 0.5 m 的距离，一般就不会有问题了。最麻烦的是通过导线进行的传导干扰，只要电子点火器和主机之间有导线存在，就会通过导线，将强大的干扰脉冲传递到 CPU 系统而造成死机，即使线缆的长度超过 20～30 m 也一样会死机。

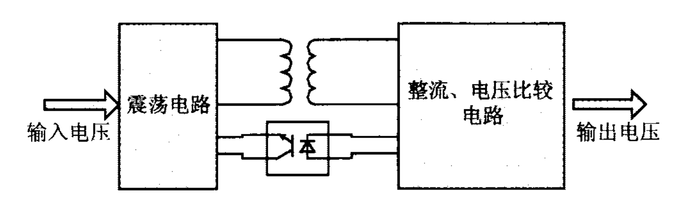


图 10.8

DC/DC 隔离电源虽然可以大幅度降低传导干扰，但是它却不能根治这种干扰。这是因为隔离 DC/DC 内部在输入电压一侧和输出电压一侧之间，一般就用开关变压器和光电耦合器来进行连接(见图 10.8)。开关变压器负责将输入一侧的能量以隔离的方式传递给输出一侧；而光电耦合器负责将输出一侧的电压误差信息传递到输入一侧的震荡电路，以调控开关变压器输送过来的能量大

小，达到稳定电压的目的。按理说，这样的隔离，应该能够彻底隔断电子点火器过来的传导干扰。可问题就在于无论是开关变压器还是光电耦合器，它们相互隔离的两侧，之间的距离很近。所以就有比较大的分布电容存在。尤其是开关变压器，它通过一层一层的线圈来传导能量，一层一层的线圈是平行排列的。分布电容尤其大。所以电子点火器的强干扰在经过分布电容后，虽然比用导线直通的方式衰减了很多，但是通过隔离层后仍然具有相当的能量，依然可以导致CPU 死机。

而像图 10.9 那样的绕线磁环就不同了，这个铁氧体磁环，它是电子电路中常用的抗干扰元件，对于高频噪声有很好的抑制作用，一般使用铁氧体材料(Mn - Zn)制成。磁环在不同的频率下有不同的阻抗特性，一般在低频时阻抗很小，当信号频率升高，磁环表现的阻抗急剧升高。使正常有用的信号很好的通过，又能很好的抑制高频干扰信号的通过，而且成本低廉。

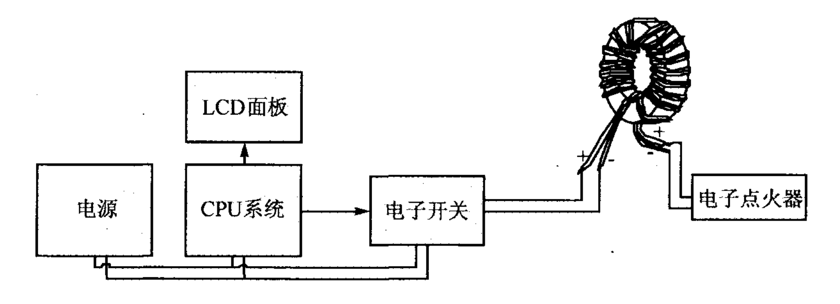


图 10.9

不知道大家有没有仔细观察 MP3 和数码相机的 USB 电缆，上面通常会有一个凸起的圆柱体(见图 10.10)。这个就是类似于图 10.9 的绕线磁环。电脑机箱内的主板、CPU、电源及 IDE 数据线都工作于很高的频率状态下，所以导致机箱里存在着大量的空间杂散电磁干扰信号，而信号强度也是机箱外的数倍至数十倍！没有磁环的 USB 线在这个空间内没有采取屏蔽措施，那么这些 USB线就成了很好的天线，接收周围环境中各种杂乱的高频信号，而这些信号叠加在本来传输的信号上，导致整机的抗干扰度下降。同时这个磁环也能防止机箱内部的强电磁辐射向外辐射而干扰其他的电子设备。

铁氧体抗干扰磁心是近几年发展起来的新型的价廉物美的干扰抑制器件，其作用相当于低通滤波器，较好地解决了电源线、信号线和连接器的高频干扰抑制问题，而且具有使用简单、方便、有效、占用空间不大等一系列优点，用铁氧体抗干扰磁心来抑制电磁干扰(EMI)是经济简便而有效的方法，已广泛应用于计

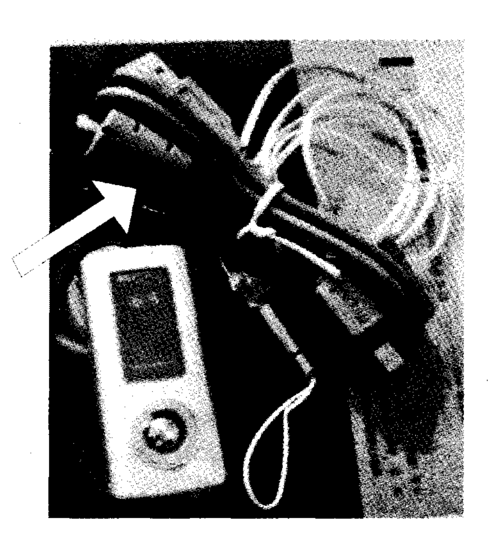
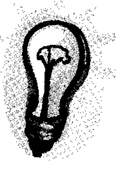


图 10.10

算机等各种军用或民用电子设备。铁氧体是一种利用高导磁性材料渗合其他一种或多种镁、锌、镍等金属在 2 000 ℃烧聚而成的。

在低频段，铁氧体抗干扰磁心呈现出非常低的感性阻抗值，不影响数据线或信号线上有用信号的传输。而在高频段，从 10 MHz 左右开始，阻抗增大，其感抗分量仍保持很小，电阻性分量却迅速增加，当有高频能量穿过磁性材料时，电阻性分量就会把这些能量转化为热能耗散掉。这样就构成了一个低通滤波器，使高频噪音信号有大的衰减，而对低频有用信号的阻抗可以忽略，不影响电路的正常工作。

将电源的两根线(正负并排)同时穿过一个磁环，有效信号为差模信号，EMI 吸收磁环/磁珠对其没有任何影响，而对于共模信号(电火花干扰脉冲)则会表现出较大的电感量。磁环的使用中还有一个较好的方法是让穿过磁环的导线反复绕几下，以增加电感量(图 10.9 就是反复绕了多圈)。铁氧体抑制元件应当安装在靠近干扰源的地方。对于输入/输出电路，应尽量靠近屏蔽壳的进、出口处。对铁氧体磁环和磁珠构成的吸收滤波器，除了应选用高磁导率的有耗材料外，还要注意它的应用场合。它们在线路中对高频成分所呈现的电阻大约是十至几百 Ω，因此它在高阻抗电路中的作用并不明显，相反，在低阻抗电路(如功率分配、电源或射频电路)中使用非常有效。



提示：

由于铁氧体可以衰减较高频同时让较低频几乎无阻碍地通过，故在 EMI 控制中得到了广泛地应用。用于 EMI 吸收的磁环/磁珠可制成各种形状，广泛应用于各种场合。如在 PCB 板上，可加在 DC/DC 模块、数据线或电源线等处。它吸收所在线路上的高频干扰信号，但却不会在系统中产生新的零极点，不会破坏系统的稳定性。

10.3 注意你的高速数字信号

某日调试一个高速的 HI－FI Audio 数字传输电路，电路如图 10.11 所示，系统由 2 个音频解码器，一个在主板上，一个在副板上。从 CPU 出来的 3 个数字信号，被分成 2 路，一路经过缓冲 IC 隔离后去 audio D/A 的第一个芯片；另外一路连接到一个接插件上，该连接器之后是一个长度为 40 cm 的一个扁平电缆，之后连接到一个单独的电路板，板上还有一个同样的音频解码器。

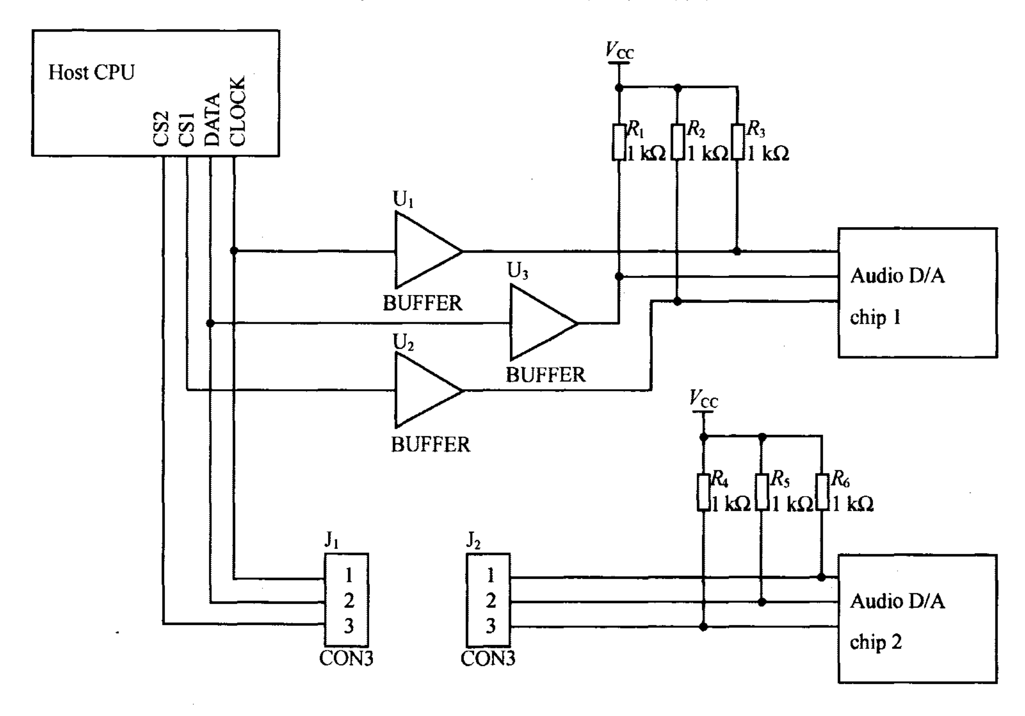


图 10.11

刚开始，只调试主电路板，一切正常，但是将连接器 J_2 和 J_1 连接在一起后，问题就来了：audio D/A 的第一个芯片会出现音量很大的杂音，而且无规律。拔掉连接器立即恢复正常。

开始以为是 J_2 连入后导致电源变化，因为小板还从大板取电，检查了电源的很多方面，如电压，纹波等，毫无异常之处。用示波器观察，发现没有连接 J_2 时的示波器图像如图 10.12 所示，连接后如图 10.13 的上半部分；经过 U_1 整形后变成如图 10.13 的下半部分的样子。

我们尝试着减小 J_2 和小电路板之间的电缆长度，发现它在长度减小到 5 cm 后，电路变得非常稳定。但是这么短的电缆在生产的时候是不能接受的，所以还要另外想办法。最后把电路更改成为图 10.14，增加了 R_4、R_5、R_6、R_7、R_8 和 R_9 六个电阻，现在再测量波形，又几乎回到了图 10.12，所以电路自然也就稳定了。

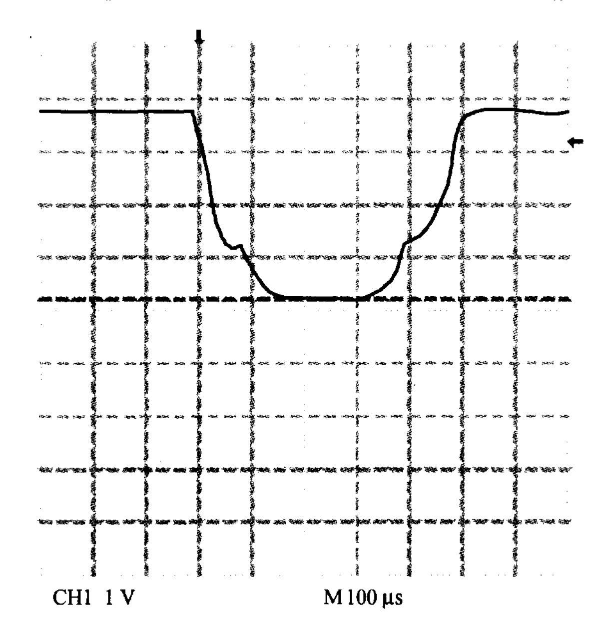


图 10.12

与以上原理类似，你可以拔一条计算机的内存条出来看看，上面除了内存颗粒，还有很多贴片的电阻（见图 10.15），而且大都靠近连接器的位置，这些基本都是降低振铃和反射等问题用的。

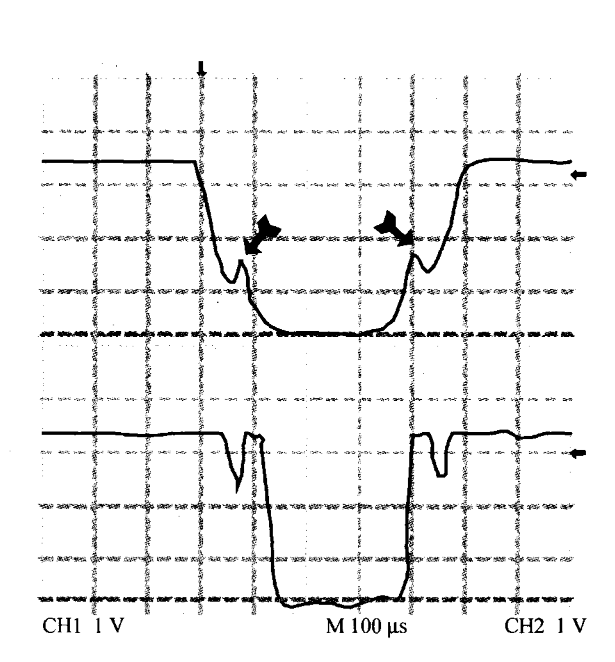

CH1 1 V
M 100 μs
CH2 1 V

图 10.13

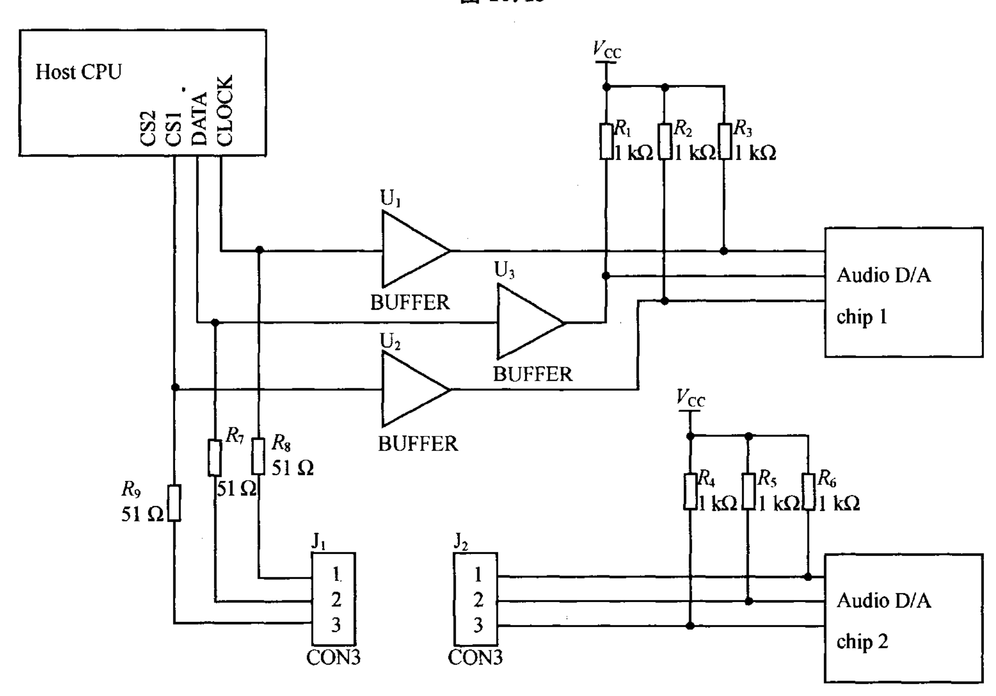

Host CPU
CS2
CS1
DATA
CLOCK
V_{CC}
R_1 1 kΩ
R_2 1 kΩ
R_3 1 kΩ
U_1
BUFFER
U_3
BUFFER
U_2
BUFFER
Audio D/A chip 1
V_{CC}
R_4 1 kΩ
R_5 1 kΩ
R_6 1 kΩ
R_7 51 Ω
R_8 51 Ω
R_9 51 Ω
J_1
1
2
3
CON3
J_2
1
2
3
CON3
Audio D/A chip 2

图 10.14



图 10.15

为什么三个电阻会解决这么大问题？



- 在这个电路里面，时钟的频率大约 800 kHz，不连接 J_2，Host CPU 输出来的信号，通过很短的路径就可以到达 chip 1，但是连接 J_2 后，由于电缆的存在，信号的通路多了一个分支，这样为信号线增加了一些不必要的分布电感和电容。而导致信号畸变。
- 由于 R_4、R_5、R_6、R_7、R_8 和 R_9 的存在，可以尽可能减小这些分布电感和电容导致的信号振铃和反射。

10.4 用电流传输替代电压传输

有关差分电路内容，请参考 8.6 节的相关内容。

LVDS 传输支持速率一般在 155 Mbps(大约为 77 MHz)以上。LVDS 是一种低摆幅的差分信号技术，它使得信号能在差分 PCB 线对或平衡电缆上以几百 Mbps 的速率传输，其低压摆幅和低电流驱动输出实现了低噪声和低功耗。IEEE 在两个标准中对 LVDS 信号进行了定义。ANSI/TIA/EIA - 644 中，推荐最大速率为 655 Mbps，理论极限速率为 1.923 Mbps。

- LVDS 信号传输一般由三部分组成：差分信号发送器和差分信号互联器和差分信号接收器(见图 10.16)。差分信号发送器：将非平衡传输的

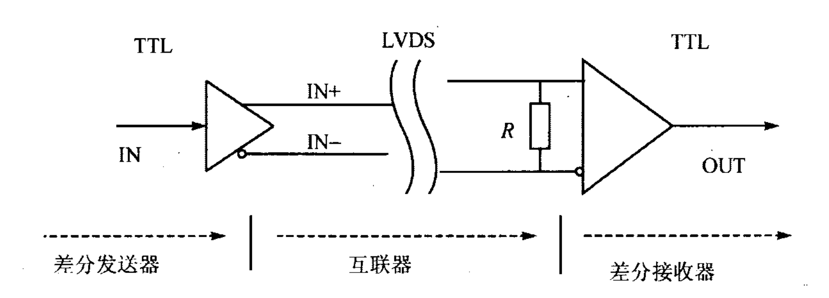


图 10.16

TTL 信号转换成平衡传输的 LVDS 信号。通常由一个 IC 来完成，如：DS90C031。

- 差分信号接收器：将平衡传输的 LVDS 信号转换成非平衡传输的 TTL 信号。通常由一个 IC 来完成，如：DS90C032。
- 差分信号互联器：包括联接线（电缆或者 PCB 走线），终端要连接匹配电阻。按照 IEEE 规定，电阻为 100 Ω。我们通常选择为 100 Ω 或 120 Ω。

正是由于使用了以上的 LVDS 差分技术，所以只需要 4～8 对差分线就可以完成 24 bit 的真彩、80～120 Hz 刷新率的高清画面的传送。这样的画面需要极高的带宽，而用普通的 TTL 电平的排线来传送，最多能做到小屏幕信号的传送，大屏幕的高清信号由于带宽太大就扛不住了。而且普通的 TTL 电平的排线由于电缆的固有特性，高带宽下电缆不可能很长。所以现在的大屏幕显示器普遍都使用 LVDS 来传送超高带宽的视频信号。

10.5 注意电源纹波

关于电源纹波干扰，我在“搞定电源”一章中已经说得很多了。这里再列举一个例子，说明注意电源纹波的重要性。

有这样一个例子，一个由 LM386 构建的音频设备，出现了一个问题：

- 当输入的音频达到一定幅度的时候，声音就开始沙哑。而小声播放音乐的时候，声音却很柔和。电路如图 10.18 所示。
- 更换 LM386 不能解决问题。
- 更换扬声器也不行。
- 失真的时候，波形如图 10.17 所示。
- 自激的时候，拔掉扬声器，用示波器观察 IC pin5，它又不自激了。

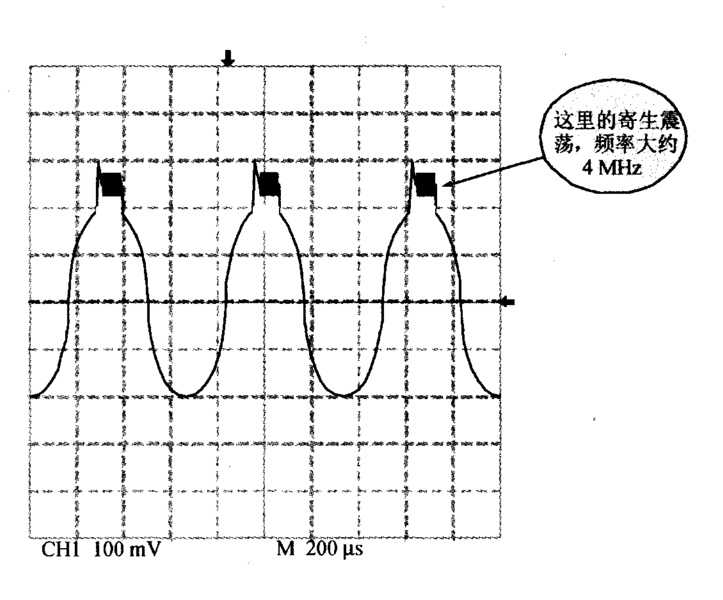


图 10.17

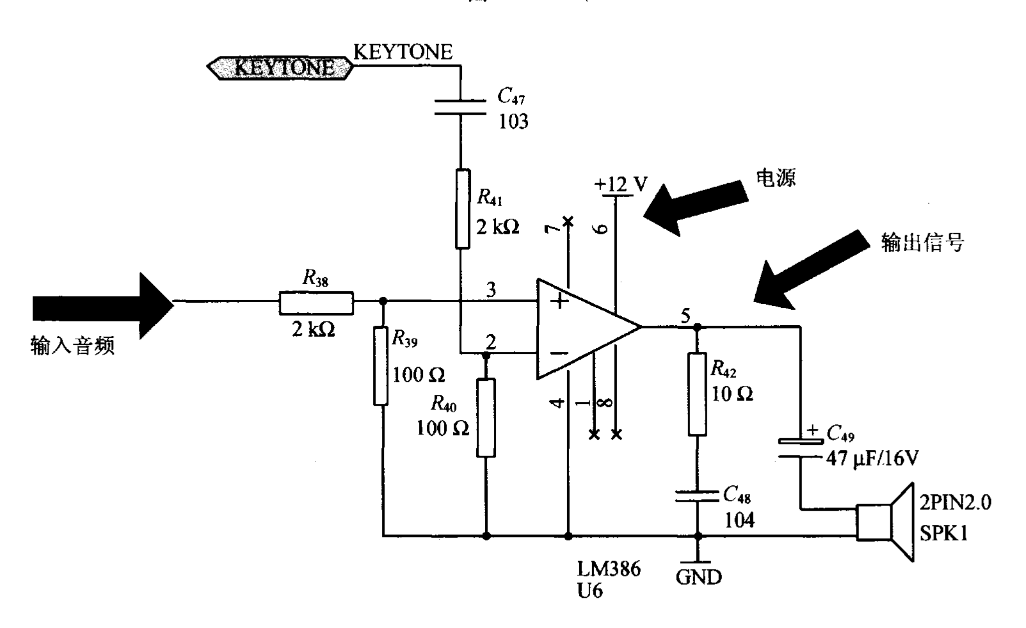


图 10.18

- 更换防止震荡的 R_{42}、C_{48}，无效。
- 当在输入端，连接一个信号发生器，给出 1.5 kHz 的正弦波信号，把信号

发生器输出信号的幅度调节到将要使 LM386 产生自激而未产生自激的时候，用手持一个镊子碰触 LM386 的 GND，也会导致自激。甚至这个时候碰触 speak 输出端或者电源端，也会导致自激。

- 但在自激临界点的时候，LM386 输出幅度，不足 1 Vpp，远不及 LM386 规格书上承诺的输出幅度。

这些故障现象，给我的感觉是，这个电路陷入了不稳定状态。但是在什么地方不稳定，却想了很久。最后，增大输入信号的幅度，让电路刚好进入稳定的自激状态后，用一个 104 电容以最短的路径焊接在 IC 的 pin6 和 pin4 之间。故障立即消失，这个时候，LM386 就能够输出 8 Vpp 的不失真功率了，用示波器看波形十分圆润，用耳朵听，声音也很柔和。

这个故障，并不是电路板上没有退偶，而是退偶的 104 电容太少，而且距离 LM386 十分遥远，退藕后经过了一段很长的铜箔才到达这个 IC，由于铜箔的阻抗和感抗的存在，削弱了远端 104 电容的退藕作用，所以留下故障隐患。而只要做到良好退偶，这种莫名其妙的干扰和失真就自己消失了。

最后讲一个案例，我在 EDN BBS 上解答网友的对话，以下是他的提问：

我的电源是用整流桥，然后用 7805 和 7905 组成正负 5 V 电源。可上面有一个 400 kHz 左右的正弦波，地线和噪声都很大，有 4 mV 左右。后面放大信号的时候都变成了大信号了。请问：

- 电源上面噪声太大，上面几百 kHz 的信号是器件的问题还是布线的问题？
- 怎么消除电源上面的噪声，电容等有什么要求吗？
- 补充问题：我处理的信号是不到 1 mV 的，在 100 kHz 左右。电源噪声太大了根本就做不了。估计要一个高性能的电源系统，做过的给点建议！

以下是我的回答：

根据楼主的要求，类似于弱信号的处理，我以前处理过 1 mV 左右的信号，截止频率为 100 Hz。按照个人的经验，在处理微弱信号时，有如下几点值得注意。

- 微弱信号的放大，如果可能的话，一定要采取共模差分输入，尽量避免采取单端输入放大。共模差分可以有效地消除共模干扰，特别是因为电源工频 50 Hz 及谐波。在选用放大器时要选用高共模抑制比的芯片。
- 电源系统采用整流桥、7805、7905 没有问题，就是在输入输出端一定要

用大电解电容和多个0.1 μF的电容并联滤波。而且要注意电路的负载电流，尽量选用冗余量大一些，这样可以避免负载过大产生电源的电压波动。其实，电源的纹波并不会影响信号放大(更何况只有4 mV)，主要是电源的工频及谐波的共模干扰。

- 电源变压器要选用中间抽头的两组输出。初级和次级间一定要有屏蔽层。不要用地摊货的变压器。
- 对于工频及谐波的干扰一般很难消除，因此，除了尽量降低电路阻抗和纹波抑制能力以外，在放大电路要采取相应的滤波电路。通常有源滤波电路比较好，如带通和陷波电路，应用电路很多，自由选取。或者采取软件数字滤波，可以考虑一个算法，来消除干扰。
- 电路布线。地线尽量环抱信号线，形成屏蔽，在有外围大电流、数字电路时，如果需要处理的信号很微弱，还要考虑用屏蔽罩。如果电路中有数字和模拟两个部分，一定要设计好模拟地和数字地的接触点，最好是单点连接，切忌形成环路。

10.6 看看无线电接收机如何抗干扰

10.6.1 几种常见的无线电接收机

1. 超再生

超再生接收机价格低廉，经济实惠，而且接收灵敏度高，但是缺点也很明显，那就是频率受温度漂移大，抗干扰能力差。正因为其抗干扰性能差，所以现在一般接收语音和视频的接收机都不使用这样的接收机。而一些对成本非常敏感的场合，比如小数据量的无线电通信的应用，比如家用电子警戒设备中，会因为其价格低廉而使用它。

2. 超外差

超外差式接收机优点是频率稳定，抗干扰能力好，和单片机配合时性能比较稳定，缺点是灵敏度比超再生低，价格远高于超再生接收机，而且近距离强信号时可能有阻塞现象。超外差式接收机利用本地产生的振荡波与输入信号混频，将输入信号频率变换为某个预定频率的电路。超外差这种方法是为了适应远程通信对高频率、弱信号接收的需要，在外差原理的基础上发展而来的。外差方法是将输入信号频率变换为音频，这种方法是将输入信号变换为超音频，所以称之

为超外差。超外差电路的典型应用是超外差接收机，其优点是：

- 容易得到足够大而且比较稳定的放大量。
- 具有较高的选择性和较好的频率特性。
- 电路调整好之后，非常稳定。
- 容易调整。缺点是电路比较复杂，同时也存在一些特殊的干扰，如像频干扰、组合频率干扰和中频干扰等。

10.6.2 超外差接收的抗干扰

简单收音机为了提高灵敏度指标增加了高放级，但高放级级数的增加是有限度的，如果为了提高灵敏度而加多个高放级，则不但统调因难，更易发生寄生振荡。另一个原因在于：半导体器件对高中低频带的放大倍数不同，这就造成了整个接收频带内的指标不和谐。

如果能把收音机固定在一个频带上工作，它的收音质量当然很好，不过事实上许多广播电台并不都挤在一个不大的频带上广播，而是分布在一个很宽的频带中进行广播。因而，只能在改进收音机的电路上想办法，把这些分散在各波段的电台，在收音机里变成一个预定的频率，这样，就能很好地加以放大了。

超外差电路就是这样的装置。它将所要收听的电台在调谐电路里第一次谐振以后，尽可能的留下 f_{in}，而将 f_{in} 以外的不需要的频率进行了第一次衰减。再经过混频电路与一个特定频率的本地震荡信号 f_o 混合，就变成另外一个预先确定好的频率 IF(一般叫做中频)，然后再进行放大和检波。这个固定的频率，是由差频的作用产生的。如果在收音机内制造一个振荡信号(通常称为本机振荡)，使它和外来高频调幅信号同时送到一个晶体管内混合，这种工作叫混频。由于晶体管的非线性作用导致混频的结果就会产生一个新的频率，这就是外差作用。采用了这种电路的收音机叫外差式收音机，混频和振荡的工作，合称变频。

变频很象货物转运。货物从遥远的地方由火车运到终点车站，然后由汽车转运到目的地。货物内容没有变，但运输工具由火车改为汽车。还可以再作简单归纳：变频仅仅是载波频率变低了，并且无论输入信号频率如何变化最终都变为固定的中频，而音频信号(包络线的形状)没变。这包络线正是我们运输的货物。

外差作用产生出来的差频，习惯上采用易于控制的一种频率，它比高频低，但比音频高，这就是常说的中间频率，简称中频。任何电台的频率，由于都变成

了相同频率的中频，放大起来就能得到相同的放大量。由于它的频率和要接受的天空信号的频率有重大差别，所以，对中频放大很高的倍数，也不至于反向辐射到接收机的天线而引起震荡。而对中频放大很高的倍数后，整机的灵敏度就会大幅度提高。

图 10.19 给出了超外差式收音机的方框图。可以看出，调谐回路的输出，进入混频级的是高频调制信号，即载波与其携带的音频信号。经过混频，输出载波的频率降低了，但音频信号的形状没有变。通常将这个过程（混频和本振的作用）叫做变频。

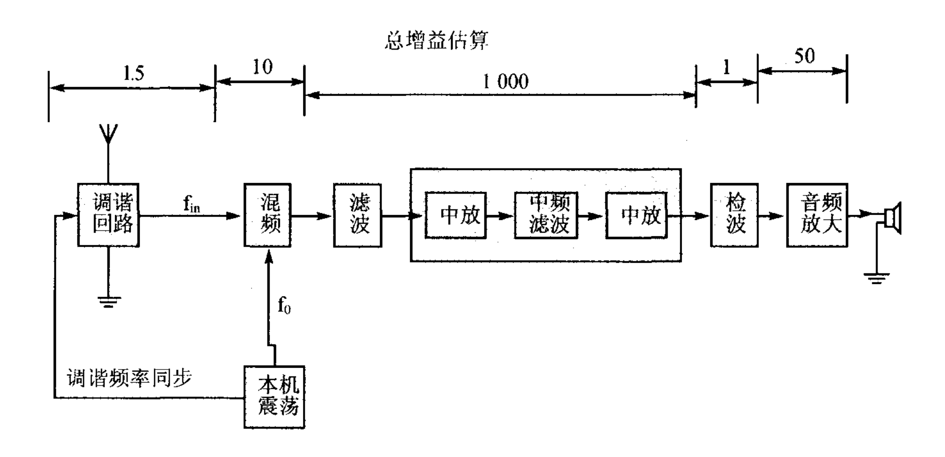


图 10.19

同样的，除了在高频信号中插入 *LC* 滤波器滤波以外，在对中频进行大倍数的放大的过程中，中间插入一个或者几个中频的滤波器，分几次对中频进行滤波，每一次都可以去掉一些不需要的信号，即噪声信号。（如图 10.20 和图 10.21的一个无线调幅 IC 电路——RX3310A）

混频器输出的携音频包络的中频信号由中频放大电路进行一级、两级甚至三级中频放大，从而使得到达二极管检波器的中频信号振幅足够大。检波电路将中频信号振幅的包络检波出来，这个包络就是需要的音频信号。若要求超外差式收音机得到更高的灵敏度，在调谐回路与混频之间还可以加入高频放大级，然后再去混频。

超外差式收音机能够大大提高收音机的增益、灵敏度和选择性。因为不管电台信号频率如何，都变成中频信号，然后都能进入中频放大级，所以对不同频率电台都能够进行均匀地放大。中放的级数可以根据要求增加或减少，更容易

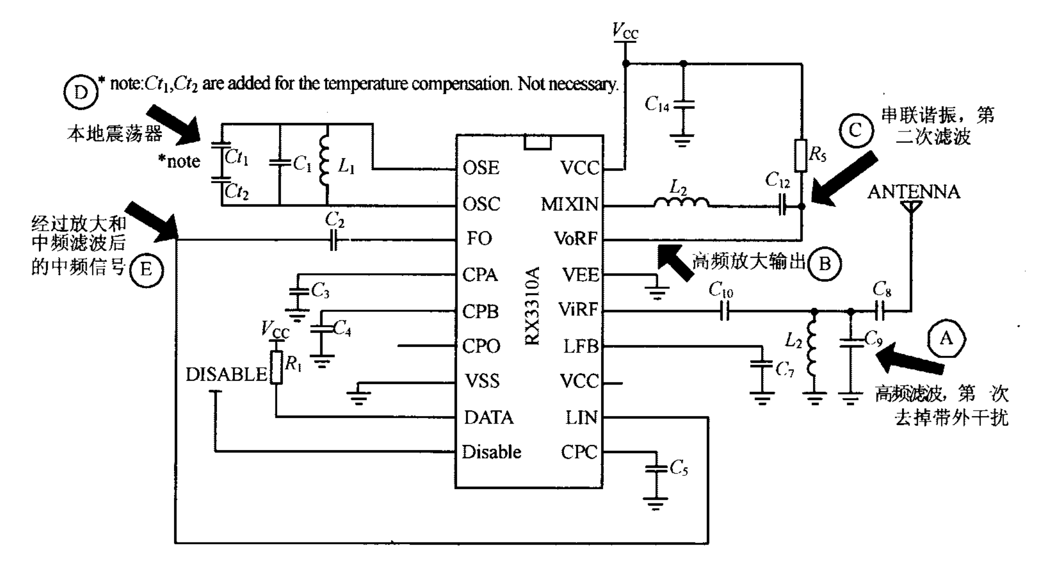


图 10.20

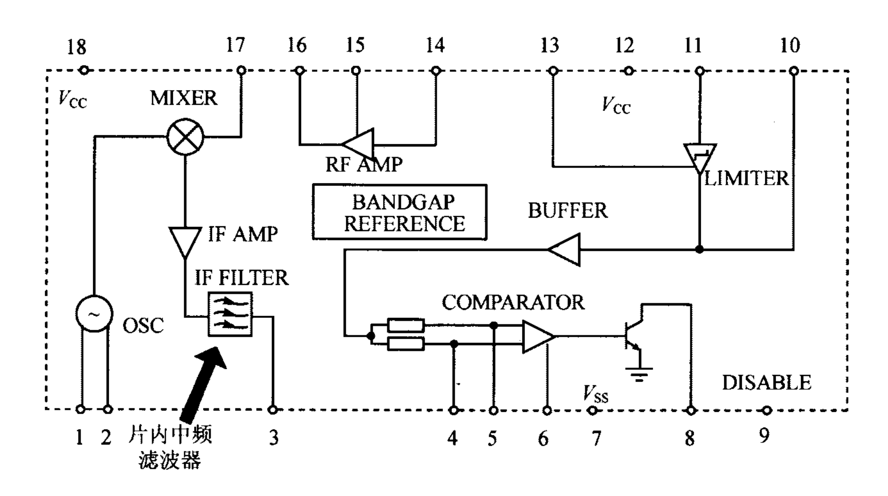


图 10.21

在稳定条件下获得高增益和窄带频响特性。此外，由于中频恒定，所以在中频放大一级不必每级都加入可变电容器选择电台，避免使用多联同轴可变电容器，而只需在调谐回路和本振回路用一只双连可变电容器就可完成选台。现在，绝大多数商品化收音机都是超外差式的。在超外差式收音机中，有时还有一些附加装置，如自动增益控制、调谐指示、负反馈和温度补偿等电路。加了这些电路，使

得收音机在质量和使用上都更趋完善，而且其中有些电路已成为不可缺少的部分。

超外差式收音机的中频放大电路采用了固定调谐电路，这一特点使它比其他收音机优越得多，综合起来有如下优点：

- 用作放大的中频，可以选择那些易于控制、有利于工作的频率(我国采用的调幅中频频率为 465 kHz，调频为 10.7 MHz)，以便得到较为稳定和最大限度的放大量。
- 各个频率的输入信号都变成了固定的中频，电路将不因外来频率的差异而影响工作，这样各个频带就能够得到均匀放大，这对于频率相差很大的高频信号(短波)来说，特别有利。
- 如果外来信号和本机振荡相差不是预定的中频，就不可能进入放大电路。因此在接收一个需要的信号时，混进来的干扰电波首先就在变频电路被剔除掉，加之中频放大电路是一个调谐好了的带有滤波性质的电路，所以收音机的选择性指标很高。去除带外干扰的能力自然也不错。

超外差式收音机和简易型收音机相比，虽然线路比较复杂，元件用的较多，因而成本较贵，但无论在灵敏度、抗干扰性和选择性等方面，都远优于简易型收音机。

10.6.3 FM VS AM

众所周知，调频接收机的信噪比要好过调幅接收机。这是因为以下 2 点：

- 调频接收机在接收的时候，信号在中频电路中被以很高的增益一直放大。直到切去中频信号的波峰和波谷。这样就免除了寄生的调幅干扰。
- 调频划分给每个电台的带宽比调幅多得多。一般中波广播的带宽一般为 9 kHz/信道。而调频广播的信道带宽要远大于这个数，达到 100 kHz 之多。所以可以说 FM 的高信噪比，其实是牺牲了带宽利用率而得到的。
- 如果压低信道带宽，仍然保持调频方式的话，那接收端的信噪比一样会降低。
- 所以降低信号带宽，而保持足够的信道带宽的话，信噪比就会变好。这个不仅在无线传输中是这样，在有线的铜缆传输中也是这样的。这也就是为什么用串口通信，波特率越低，传输的距离就越长的道理。

10.7 积分法去除噪声

不知道大家有没有忘记高等数学里面的微积分。在这里不想罗列一大堆的数学公式,只讲讲我对微积分的感受:

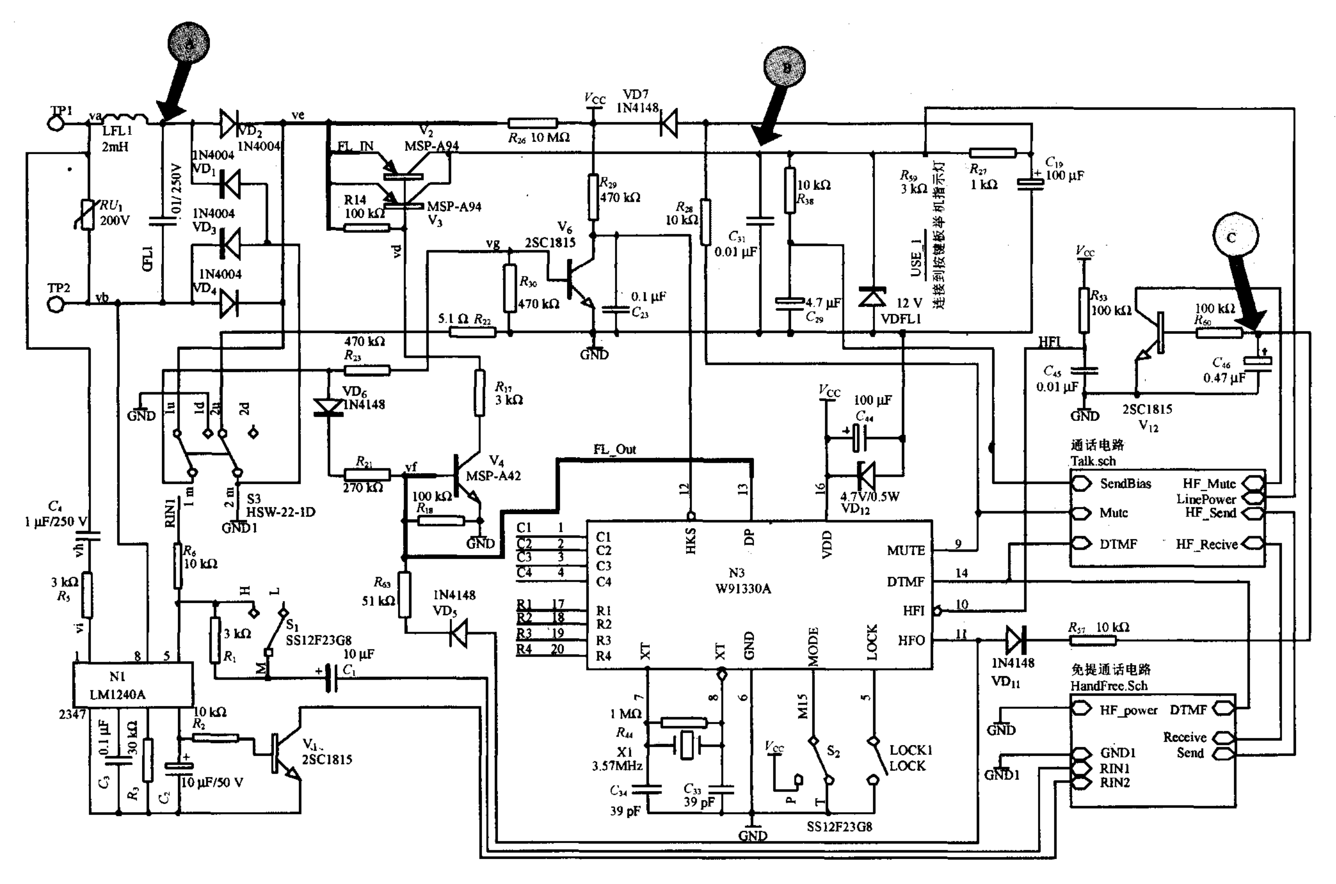
- 微分在电路里面的作用就是尽可能地提取出短时间内脱离信号平均值的信号成分。留下的信号都是偏向高频的信号。
- 积分刚好相反,在电路里面的作用就是尽可能地去掉短时间内脱离信号平均值的信号成分。留下的信号都是偏向低频的信号。
- 因此,微分电路多用于视频电路里面的图像边缘增强等,需要拉升高频的电路。但是这个电路在增强图像边缘的同时,一般也会增强图像的噪声。因为它们都是高频信号。
- 积分电路多用于去掉电路里面的高频噪声。比如几乎所有的驻极体麦克风两端都会并联一个100～100 000 pF的电容,就是为了去掉麦克风接收的高频杂音。

图10.22是一个电话机的电路图,在图中标示了A、B和C三个点。

A点的一个0.01 μF电容的作用有2方面:一个是和LFL_1、RU_1共同构成一个防雷击电路。由于电容的积分作用,可以大大缓解雷击的浪涌电流,保护内部半导体器件的安全。同时它还可以滤除电话线上的一些话音频带之外的干扰;B点的一个0.01 μF电容利用自己的容抗和V_2、V_3、VD_1～VD_4、LFL1的阻抗和线路的阻抗构成一个*RC*积分器,进一步滤除电话线上的一些话音频带之外的干扰。

C点上的电容,不是用于滤除话音干扰的,而是防止电话免提被打开的瞬间,电话机会“噗”的一声。因为如果这个电容不存在的话,打开免提后,由于免提电路瞬间上电,电路肯定会产生“噗”的一声,而加上C_{46}后,打开免提电路后,由于电容两端的电压不可瞬变,电路先给出HF_MUTE信号。等C_{46}的电压上升,导致V_{12}饱和后,电路的MUTE信号才去掉,开始正常通话。C_{46}和R_{57}一起构成了一个积分器,完成了一个积分的延时过程。

从前看一个Discovery channel的探案故事,讲一个人,夜间作案后驾车潜逃。由于夜间灯光昏暗,作案现场的摄像机只拍下了图像质量很差的视频。而且由于噪声太大图像麻点太多,车牌的号码根本看不出来。为了便于描述这个问题,我在Google上找了一个车牌,暂且充作警方要找的这个车牌(车牌的主人



TP1
va
LFL1
2mH
ve
VD2
1N4004
VD1
VD3
VD4
RU1
200V
.01/250V
CFL1
TP2
vb
FL_IN
V2
MSP-A94
R14
100 kΩ
V3
vd
Vcc
VD7
1N4148
R26 10 MΩ
R29
470 kΩ
V6
2SC1815
vg
R30
470 kΩ
0.1 μF
C23
GND
R28
10 kΩ
C31
0.01 μF
10 kΩ
R38
4.7 μF
C29
12 V
VDFL1
USE_1
连接到按键板举机指示灯
R59
3 kΩ
R27
1 kΩ
C19
100 μF
5.1 Ω R22
470 kΩ
R23
VD6
1N4148
R17
3 kΩ
R21
270 kΩ
vf
V4
MSP-A42
100 kΩ
R18
S3
HSW-22-1D
GND1
RIN1
R6
10 kΩ
C4
1 μF/250 V
vh
3 kΩ
R5
vi
R1
S1
SS12F23G8
10 μF
C1
N1
LM1240A
2347
10 kΩ
R2
V1
2SC1815
C3
0.1 μF
R3
30 kΩ
C2
10 μF/50 V
R63
51 kΩ
VD5
1N4148
FL_Out
C1
C2
C3
C4
R1
R2
R3
R4
HKS
DP
VDD
MUTE
DTMF
HFI
HFO
N3
W91330A
XT
GND
MODE
LOCK
1 MΩ
R44
X1
3.57MHz
C34
39 pF
C33
39 pF
M15
S2
LOCK1
LOCK
SS12F23G8
100 μF
C44
4.7V/0.5W
VD12
R53
100 kΩ
HFI
C45
0.01 μF
2SC1815
V12
100 kΩ
R60
C46
0.47 μF
通话电路
Talk.sch
SendBias
Mute
DTMF
HF_Mute
LinePower
HF_Send
HF_Recive
1N4148
VD11
R57 10 kΩ
免提通话电路
HandFree.Sch
HF_power
GND1
RIN1
RIN2
Receive
Send

图10.22

看见这个文章不要用砖头扔我!)。为便于说明问题,我用 Photoshop 作了一个类似的效果,正常的车牌照片如图 10.23 所示,有一点噪声的车牌如图 10.24 所示,有严重噪声的车牌如图 10.25 所示。而这个故事中的车牌的视频画面质量大概还不如图 10.25 所示。但是警察们找到一个工程师,工程师想出一个办法:



图 10.23



图 10.24

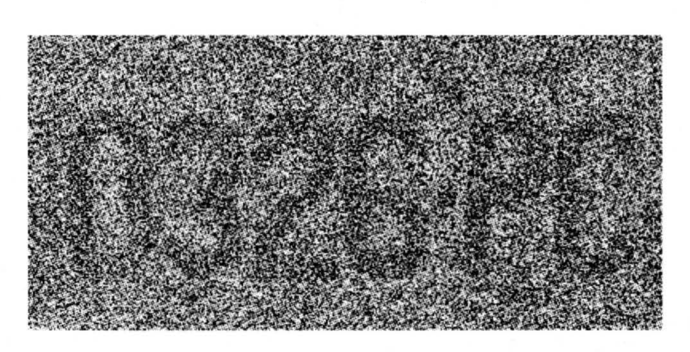


图 10.25

- 既然是视频，一定有很多包含近似画面的单独的帧。
- 每个单独的帧做成一个图片。
- 每个图片的车牌部分用 Photoshop 单独抠图出来，做成统一的 M * N 像素的图片。
- 然后用软件处理这些车牌图片，分别把一张一张图片上的相同坐标的像素的亮度值累加起来，放在一个 M * N 的 2 维矩阵里面。即：

  2 维矩阵像素点[x,y]=图片 1[x,y]点的亮度值 +图片 2[x,y]点的亮度值 +图片 3[x,y]点的亮度值 +…+图片 n[x,y]点的亮度值
- 之后把这个 2 维矩阵里面的每个元素除以图片的张数，得到一个新的 M * N 的图片。
- 这个最后生成的 M * N 的图片就是一个有很高信噪比的图片，轻易的就可以看出车牌号码。

这样处理后，大致相当于按照图 10.26 的样子，把打印有车牌号的一大堆透明菲林片，准确定位后叠合在一起，透光看的效果。

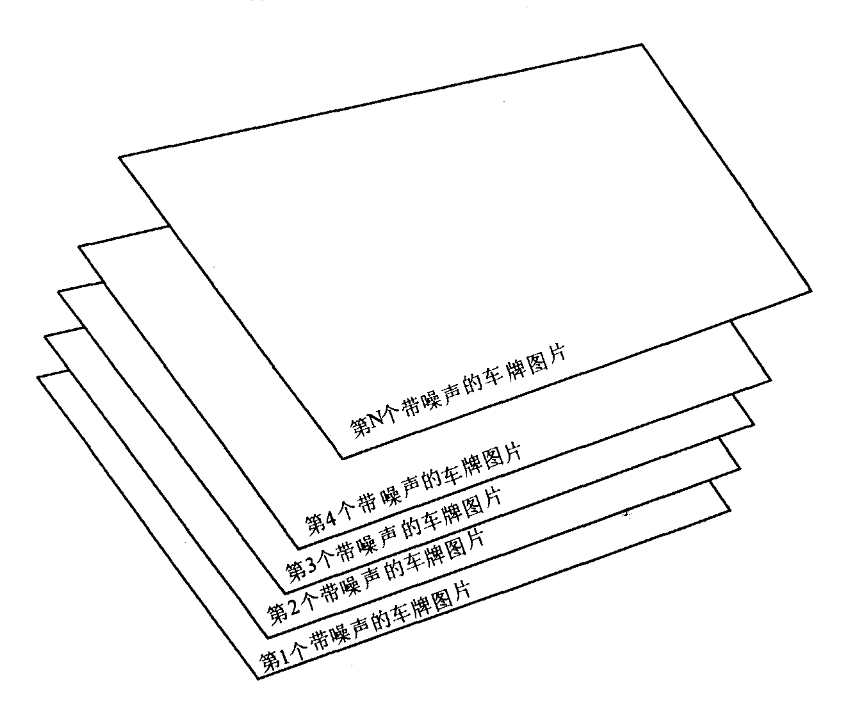


图 10.26

为什么这样处理就能去掉噪声呢？

- 因为，在每个画面里面，如果没有噪声的话，应该每个画面的对应点的亮度都差不多。
- 摄像机在弱光下体现的噪声是画面上的麻点，是一个随机信号，而且以高频成分居多。
- 在每个画面里面，对应点的噪声信号的亮度也是随机的，因为在第一副画面的噪声可以是正值，第二副画面就可能是负值。所以，如果所有的噪声累加起来，噪声的累加和虽然不一定会等于 0，但是一般会趋向于 0。
- 所以正经的信号经过多次累加后会越来越强，而噪声经过多次累加后会越来越弱。
- 这样对随机噪声的一次次的累加过程，实际上就是一种积分运算。

同样的原理，如果你用看图软件 ACDSEE 打开一个曝光不足的数码相机照片，以 100％的比例看一个照片的话，有可能看见这个照片有比较多的麻点，这就是 CCD 光学传感器的噪声，而你如果用 50％或者 25％的比例观赏这些照片的话，画面就会纯净得多。用术语说的话，那就是：“照片缩小 50％，信噪比好多了”。这里之所以画面的信噪比会变好，就是因为照片缩小 50％显示，实际是将真实画面上的 4 个点（X 轴 2 个、Y 轴 2 个，2 * 2＝4）取平均值后显示，这样的话，信号没有多少真正的损失，但是噪声却经过 4 个点的平均而弱化，如果你对照片的噪声比较敏感的话，这样的变化可以用肉眼很容易地看出来。

11

搞定电源

一般人可能会觉得电源大概是电子设备里面比较容易搞定的门类，因为，只要线路没有接错，电源都能工作。从这个方面说，好像是很容易。但是通过我多年的经验，发现电源其实是一个很麻烦的东西，稍微有一点不完美就会有大问题出现。所以专门拿一章来讲讲“最简单”的电源。

11.1 电路板上的电源就相当于人体的循环系统

如果将人体也当作是一个硬件设备，我觉得电源在一个电子系统中的作用，十分类似于人体的循环系统，因为：

- 它们都是为系统的每一个模块、每一个最小单元输送能量的。电路里面的最小单元可以是一个三极管或一个电阻；人体的最小单元就是一个个的细胞。
- 它们都是封闭的回路。
- 循环系统的血管，从心脏出来后，是一级一级逐渐变细，滋润每个细胞，然后再一级一级汇合起来，返回心脏。电路板上的电源也是这样，首先从正极出来，然后分散到每一个电路里面的最小单元，最后汇总回到电源的负极。
- 心脏为血液提供一定的压力，以保证循环的畅通。血压太高太低都会有麻烦。电源也有一定的电压，电压太高太低也同样会有大麻烦。
- 电路的功率越大，电流也越大。人体需要的能量越大，血液的流量和流速也越快。
- 血压过高或者过低可以让人毙命。而稍微有点高或者稍微有点低，就会有很多稀奇古怪的疾病发生。
- 同样的，电压过高可以让电路板立刻冒烟，电压过低可以让电路板立即罢工。而电压稍微有点高或者稍微有点低，也同样会有很多稀奇古怪的

故障发生。

- 身体供血不稳定，可以造成很多急性或者慢性疾病，比如心绞痛、紫绀等。身体供血不稳定就相当于电源系统的脉动纹波。
- 如果人体的某一个器官沾染了有毒的东西，很容易通过循环系统扩散到全身。而这个毒性超过了人体正常的代谢能力，就会有急性或者慢性的中毒症状。
- 如果电路上一个模块产生了比较强的噪声，也很容易通过电源回路干扰到电路板上的其他单元。如果其他单元的抗噪声能力比较好，那出现问题的概率就会低一些，否则就极易体现出故障。

要控制电源电压不要太高或者太低，比较容易，这里就不叙述了。本章着重介绍如何控制电源的其他方面，以保证电路板上能量系统的稳定性。

11.2 电源一定要留足余量

电源一定要留足余量，一般要比负载的峰值耗电至少多 20%，这样比较安全。电源余量不足会发生很多意想不到的故障，以下是我见过的由此而产生的故障：

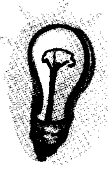
- CPU 无故复位，CPU 在不同模式下，其耗电不一样。同样一个 CPU，它的耗电上下可以差距 100 倍之多。如果电源没有满足 CPU 的峰值消耗，那 CPU 满负荷工作的时候，就是噩梦的开始，比如死机或电路板上某个单元不工作。这种情况在 32 位嵌入式系统上发生的比较多。因为那些 CPU 动辄就要好几瓦、甚至超过 10 W 的峰值功率。而平时也就几十或上百 mW。
- 一个音频功率放大电路，在重低音输出的时候，声音总是失真（比如低音大鼓敲响的时候，声音发闷）。而平时的人声，却婉约动人。因为重低音输出的时候，是功率放大器最耗电的时候。电源的功率储备不够，就会影响音质。
- 一个嵌入式系统，原先都工作的很正常。某一天突然以太网部分失灵。经过检查，是因为电路板上多增加了一个视频编解码芯片，导致主电源耗电大增 200 mA。这个时候电压从 3.3 V 跌落到 2.9 V。而 CPU 依然正常工作，可是以太网芯片却因为这 0.4 V 的电压跌落而罢工，而

CPU 和视频编解码芯片却可以正常运转。但是和以太网芯片通信的时候，因为以太网芯片已经处于不稳定状态，所以就不正常了。重新处理电源后恢复正常。

通常电源余量不足，电源会工作在极限状态，电源的纹波会剧烈上升。达到几百 mV 甚至几 V。同时电压也会下跌。有些开关电源更加恶劣，达到了它的极限功率，它就关闭所有输出，让负载彻底断电，然后等待重启这个电源。

如果电源的容量只差一点点呢？那样可能电路板上的电压也会下降一点点，这样，电路板出现故障的概率也增加一点点。不要小看了这一点点，这个可能意味着电路板量产时，每 1000 片电路板就会有几片、几十片出现莫名其妙的毛病，虽然总数不多，可是一样能把设计的责任工程师整趴下。

爆发力好的运动员，除了有一套爆发力很好的肌肉以外，还需要一套能适应爆发力的循环系统。同样的，电源是电路板的能量系统，没有充足的能量，电路板还能好好工作吗？如果需要飞机能突破音障，就必须为飞机安装一个动力强劲的涡轮喷射引擎，无法想象，缺乏爆发力的活塞引擎也能超音速飞行。



提示：

● 如何有效评估一个系统的电源需求。

对于一个实际的电子系统，要认真分析它的电源需求。不仅仅是关心输入电压、输出电压和电流，还要仔细考虑总功耗，电源实现的效率，电源部分对负载变化的瞬态响应能力，关键器件对电源波动的容忍范围以及相应允许的电源纹波，还有散热问题等等。功耗和效率密切相关，效率高了，在负载功耗相同的情况下总功耗就少，对于整个系统的功率预算就非常有利，对比 LDO 和开关电源，开关电源的效率要高一些。同时，评估效率不仅仅要看在满负载时候的电源电路的效率，还要关注轻负载时候的效率水平。

至于负载瞬态响应能力，对于一些高性能的 CPU 应用会有严格的要求，因为当 CPU 突然开始运行繁重的任务时，需要的启动电流很大，如果电源电路响应速度不够，会造成瞬间电压下降过多、过低，造成 CPU 运行出错。

散热问题对于那些大电流电源和 LDO 来说比较重要，通过计算也可以评估是否合适。不过，如果是大电流应用，最好不要用 LDO 之类的线性稳压器。

- 如何选择合适的电源实现电路。
  根据分析系统需求得出的具体技术指标，可以来选择合适的电源实现电路。一般对于弱电部分，包括了 LDO（线性电源转换器），开关电源。相比之下，LDO 设计最易实现，输出纹波小，但缺点是效率有可能不高，发热量大，可提供的电流相较开关电源不大等等。而开关电源电路设计灵活，效率高，但纹波大，实现比较复杂，调试比较烦琐等等。

- 关于电源部分的布线，要注意如下几个方面：
  1. 两条线环路过大。
  2. 电气点的线路过长。
  3. 功率线路的铜箔宽度宽一些、连接电缆也尽量粗一些。
  4. 功率器件及干扰源器件要注意摆放的位置和方向，否则也会使电路板上的敏感部位受到干扰。
  5. 退耦电容的容量不足，或者在电路板上安排的位置不合理。

11.3 纹波所带来的一系列麻烦

理想的直流电源输出的电源应该是纯正的直流，没有一丝一毫的杂波。但是，正因为这种电源叫做"理想的直流电源"，所以它除了出现在书本上，在现实生活中是不存在的。现实的直流电源，无论其种类，总会有内阻。所以它在给各种负载供电的时候，负载的电流变化，就会在电源上以纹波噪声的形式体现出来。而且有些电源本身就会有输出的脉动，这同样也是纹波。

所谓纹波就是叠加在直流电源上的小幅度的交流信号。纯正的直流，输出电压等于一个常数 C；而有纹波的电压，它的输出表示成数学公式，如式 11－1 所示，其中 sin(A)、sin(B)、sin(C)和 sin(N)表示各种频率的信号。a、b、c 和 n 表示纹波中各个对应频率的衰减系数。C 后面的式子，其实就是纹波信号的傅

立叶展开式。

$$V = C + \frac{\sin(A)}{a} + \frac{\sin(B)}{b} + \frac{\sin(C)}{c} + \cdots + \frac{\sin(N)}{n} \quad （式 11-1）$$

在设计工作中发现，电源纹波会导致很多故障的发生。故障一般比较隐蔽和随机，多以难处理的软故障形式出现。正是因为电源变得如式11-1，在上面叠加了这些形形色色的、不可预测幅度和频率的交流信号，才会呈现出各种稀奇古怪的症状。比如：

- 音频系统中，扬声器的交流声或者其他频率的杂音。
- 视频系统中，图像有条纹。
- CPU的死机，很多硬件工程师设计的电路板不合格。导致软件频繁死机，害得软件工程师没日没夜的加班，却不能解决问题。
- AD转换器的转换结果不准确。
- 电路板上某个模块失灵。

电源线上的噪声或纹波的来源具有多样性。一般是因为以下原因导致的：

- 电源容量不足，非常容易导致纹波电压快速大幅度上升。
- 在系统内的高速数据和高频信号本身会产生噪声，然后反向影响到电源部分。
- PCB的印制线和连接线如果设计不当，电阻增大而增大了纹波。
- 数字IC，例如微控制器和现场可编程门阵列（FPGA）以及复杂可编程逻辑器件（CPLD）在高速运行的时候，具有很快的边沿跳变速度，瞬时电流会有大幅度变化，将产生电磁干扰串扰和辐射到系统中，最容易产生的干扰就是通过电源向外辐射而干扰其他部件。如果处理不当的话，会比电源产生的纹波更加难以处理。这个很类似于某人被蛇咬到了手，蛇毒通过循环系统而污染了其他脏器。

11.4 如何镇压电源的纹波？

使用LDO可以有效地降低纹波，因为LDO工作于完全的负反馈方式，所以它的电源抑制比（PSRR）较开关电源要优秀得多。所以，如果在小信号的音频、视频、CPU电路前面如图11.1和11.2所示用LDO串联供电，都能取得不错的效果。在信号回路使用LDO要注意以下问题：

- LDO电路的输入和输出需要对地并联电解电容器。通常在100 μF以上。较低内阻的大电容器一般可以全面提高电源抑制比（PSRR）、噪声

以及瞬态性能。

- 如果信号回路上有寄生的低频振荡,(比如音频电路中,喇叭里面有“噗噗噗”的周期性声音)一般是电源的低频内阻太大,增加对地并联的电解电容器的容量,通常可以解决。
- 在对地并联的电解电容器旁边一般还要并联 0.1 μF 的陶瓷电容,以保证电源的高频响应能力。
- 如果信号回路上有寄生的高频振荡,(比如音频电路中,喇叭里面有“吱吱”的尖叫声、视频电路上图像有垂直条纹)一般是电源的高频内阻太大,增加对地并联的陶瓷电容的容量,通常可以解决。
- LDO 如果是给信号回路供电,那这个 LDO 就不要再给 CPU 之类的数字电路供电。否则数字电路跑起来后,会污染 LDO 所输出的纯净电源。

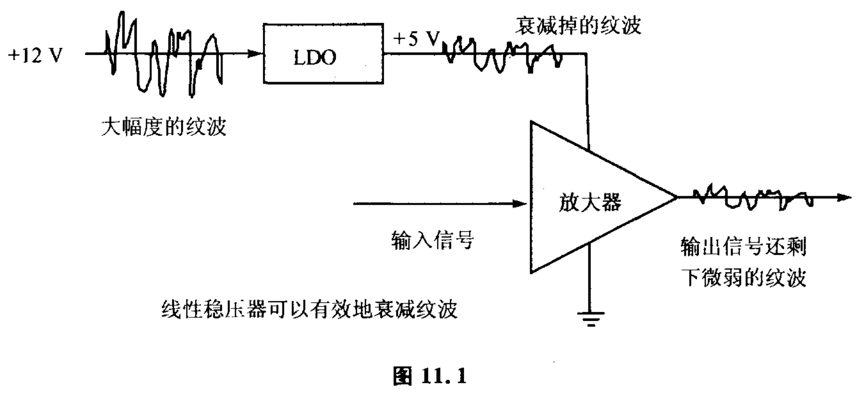


图 11.1

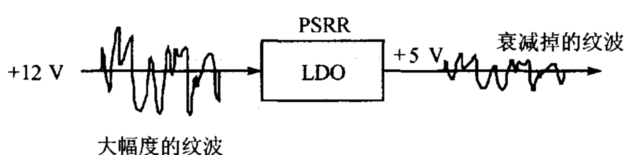


图 11.2

有三种常用的方法来使信号路径中的噪声和纹波最小:非常仔细的系统 PCB 布局、恰当的电源旁路处理以及正确的电源选择。尽管 PCB 的具体设计取决于系统,但就一般而言,PCB 的布局需要考虑包括正确的器件布局、使信号路径连接线的长度最小以及采用实体的地等。

在作电源旁路的时候，陶瓷电容器通常是旁路高频的首选，因为它们价格低而且故障模式是断路，相比之下钽电容器比较昂贵且其故障模式是短路。输出电容器的等效串联电阻(ESR)会影响其稳定性，陶瓷电容器具有较低的 ESR，大概为 10 mΩ 量级，而钽电容器 ESR 在 100 mΩ 量级。但钽电容由于其很容易做到 10 μF 以上的容量，所以，在 DC/DC 和开关电源中用得也比较多。

如图 11.3 和图 11.4 所示，我们来看看一个电脑主机板的电源部分的安排(台式机 CPU 的耗电很大，它的电源设计比较说明问题)：

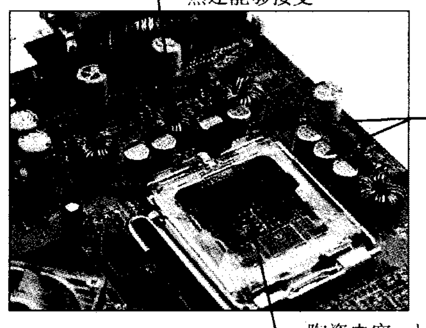


图 11.3

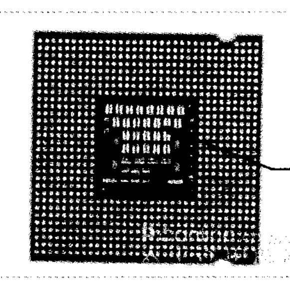


图 11.4

- 降低纹波的幅度就是大大提高式 11-1 中的 a、b、c 及 n 等小写字母的数量级，将各个没用的频率成分尽可能的衰减掉。
- 电解电容负责衰减频率较低的纹波，因为频率低，所以电解电容在电路板上的安放位置并不十分重要。只要在电路板上，一般效果都差不多。

- 钽电容负责衰减频率中等的纹波。
- 陶瓷电容负责衰减频率较高的纹波，因为高频对线路的传输效应明显，所以它要尽可能的靠近会产生噪声的地方，比如芯片的管脚焊接处。

11.5 如何尽可能的容忍电源的纹波噪声

稍微资深一点的硬件工程师都知道，DC/DC 电源电路，虽然通电后电路板会很凉快，但是它也有很讨厌的地方，就是电源的纹波一般比较大。DC/DC 开关电源的纹波噪声原因很多，最主要的是斩波频率造成的，所以在选择 DC/DC 芯片的时候要尽可能选择频率较高的，频率高的好处有：

- 频率高，其纹波的频率也就高，这样的纹波也更容易滤除。
- 频率高，就可以选低感值的电感，这样就有更强的负载能力。
- 频率高，在负载不是很大的情况下，可以实现用小的电容实现理想的滤波效果。

- 频率高的缺点是自损耗电流大。

在设计 DC/DC 开关电源的时候，我比较喜欢使用非常便宜的芯片 MC34063，这个芯片的批量采购价格不足 0.8 元人民币。它的默认工作频率为 50 kHz，对于一些数字电路的应用场合，没有任何问题，但是在为一些模拟电路供电的时候，麻烦就来了，主要体现在：

1. MC34063 供电的音频电路里面总是会有很烦人的“嘶嘶”噪声，类似白噪声的感觉。

2. 开关频率过低，导致纹波难以下降。（便宜芯片的臭毛病，频率高、纹波小的芯片，其价格可以抵得过好几片 MC34063）。

电源的波动影响模拟电路，主要有以下几个途径：

1. 开关噪声的空间辐射。

2. 电源纹波通过模拟器件的电源端窜入信号回路，经过层层放大，最后达到可以听见的程度。

3. 通过电路的参考电压或者偏置电压窜入信号回路。

以我个人的经验而言，以上三条，对电路造成的影响严重程度，是：途径 3>途径 2>途径 1。如果就发生的概率来说，则是：途径 2>途径 1>途径 3。

开关噪声的空间辐射，在我所设计过的电路中，感觉影响并不算很大。只要 DC/DC 电源部分和比较敏感的模拟部分，有足够的距离，一般没有问题。如果

是非常微弱的模拟信号，一般就不要使用 DC/DC 电源，或者距离它更远一些，或再加一个屏蔽罩，一般都可以解决空间辐射的问题。

电源纹波通过模拟器件的电源端窜入信号回路，这种情况一般在三极管做的分立器件电路里面比较严重，因为三极管做的分立器件电路，它们的偏置电路一般都是通过一个到 2 个电阻直接连接电源（见图 11.5），电源的电压有微小的变化，经过三极管的放大后，噪声就有可能会达到不可接受的地步。改进的方法如图 11.6 所示。

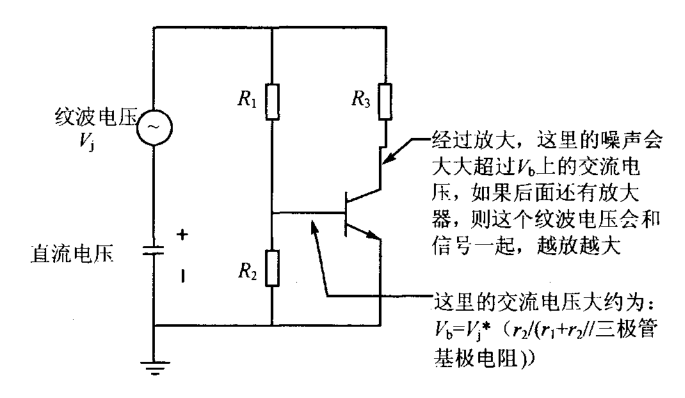


图 11.5

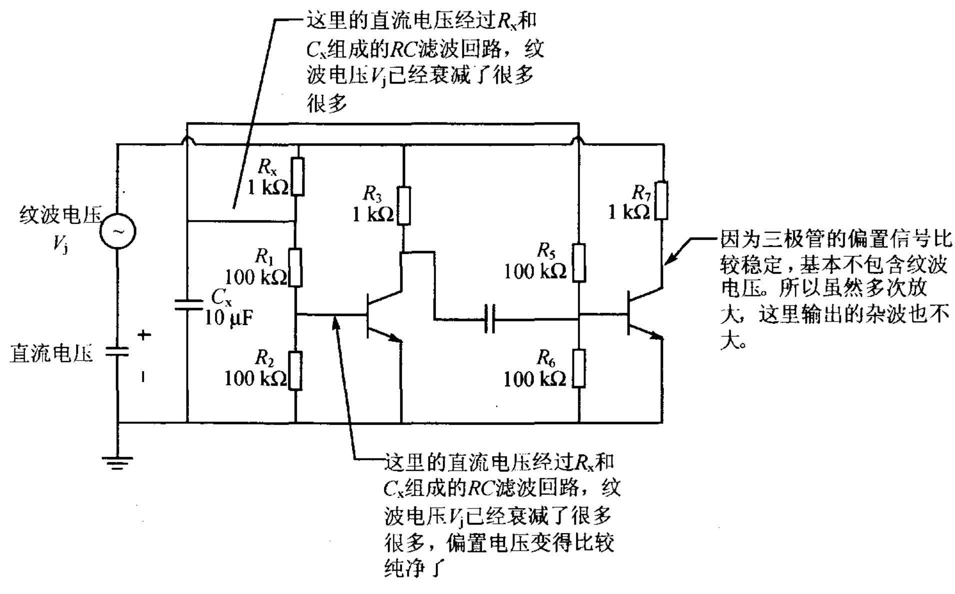


图 11.6

电源纹波通过模拟器件的电源端窜入信号回路，这种情况在共模传输的信号回路一般比较严重，而三极管做的分立器件电路因为要控制成本，通常都是设计成共模传输的信号回路。不过这也是可以解决的。图 11.6 的改进，已经可以减少很多输出的纹波噪声，但是还会有一些噪声由三极管的 C 极窜入。这个时候可以把电路更改成类似于图 11.7 的样子。为三极管的 C 极供电再来一次 *RC* 滤波。这样处理后，你在输出端想得到噪声都很难。

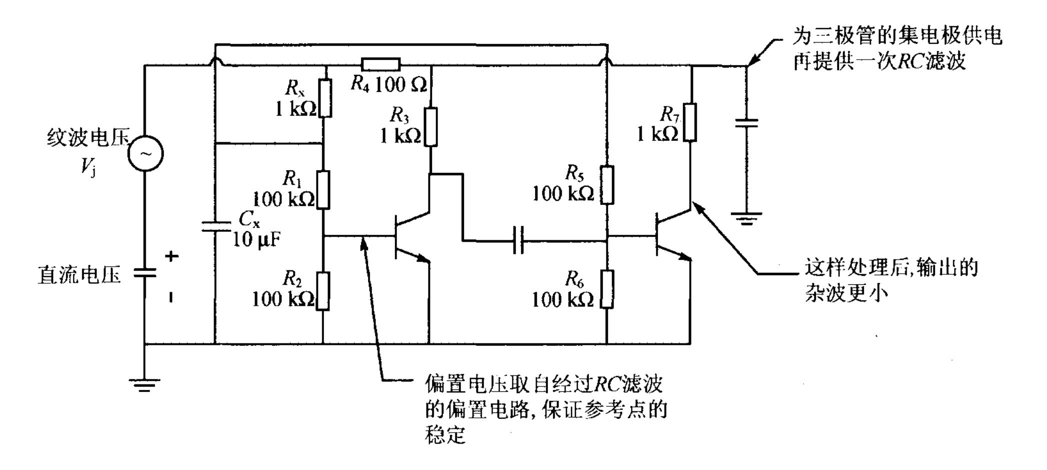


图 11.7

在这两点上，通常 IC 做的电路要好得多，原因大致有以下几条：

- IC 做的电路，其内部经常是差分的信号传送形式。这种形式本身抗干扰性能就不错。
- 很多 IC，都为其应用场合做了特别的带宽限制，限制住了工作带宽，自然不会受到带宽外的信号干扰。因为距离器件的限制带宽越远，其放大系数越小。比如图 11.8，它是一个音频运算放大器的内部电路图，图中圆圈中的电容，就是用于限制工作带宽的。加入这个电容后，这个运算放大器的带宽就限制在 0～500 kHz 以内。而三极管的频率响应，动辄都有 50～100 MHz，如果需要限制带宽，一般需要另外增加元件。电源纹波中，经常也有这样高频率的噪声。所以，如果有用信号中没有这么高的频率的话，用低带宽的器件反而会更稳定。图 11.9 是 LM358 音频运算放大器的电参数。
- IC 内部电路的偏置，大多通过恒流源或者恒压源来提供，无论是恒流源还是恒压源，都可以大幅度衰减电源纹波窜入内部电路的偏置，保证了



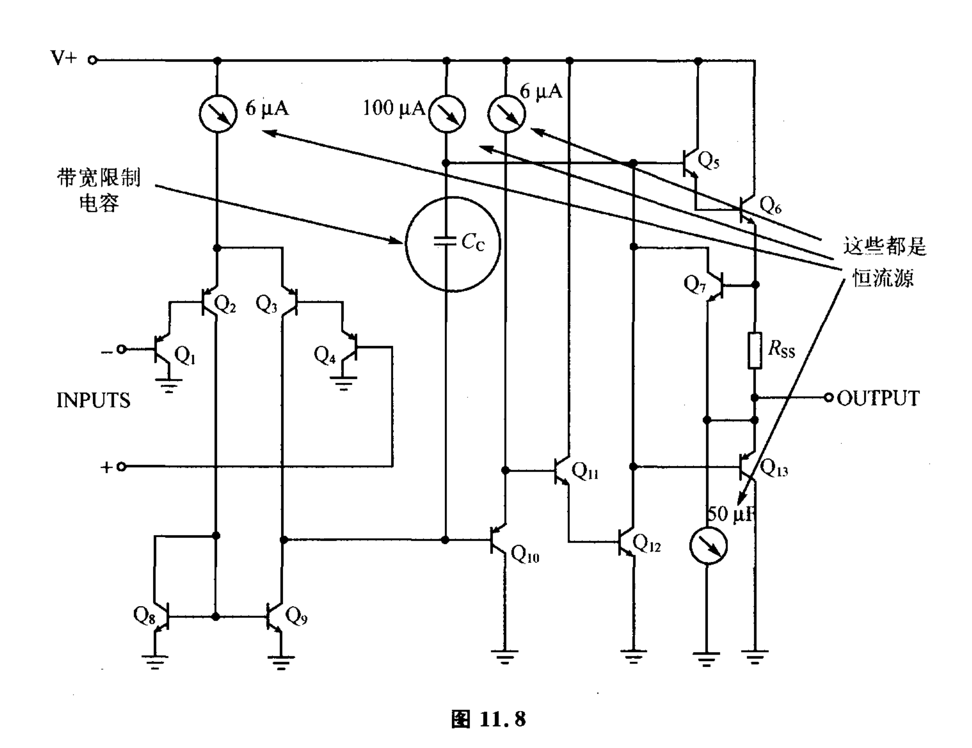


图 11.8

Electrical characterisitics at specified free-air temperature, V_{CC}=5 V(unless otherwise noted)

PAPAMETER	TEST CONDITIONS*		LM358 MIN	LM358 TYP	LM358 MAX	UNIT
CMRR Common-Mode Reiection Ratio	V_{CC}=5 V to MAX, $V_{IC}=V_{ICR}$ MIN	25 ℃	65	80		dB
K_{SVR} Supply Voltage Rejection Ratio($\Delta V_{CC}/\Delta V_{IO}$)	V_{CC}=5 V to MAX	25 ℃	65	100		dB
V_{01}/V_{02} Crosstalk Attenuation	f=1 kHz to 20 kHz	25 ℃		120		dB

因为是差分电路而带来的很高的共模抑制比

因为是差分电路和恒流偏置而带来的很高的电源纹波抑制比

图 11.9

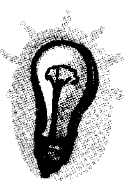
电路的稳定性。恒流源有一个特性：理想的恒流源它的交流阻抗无穷大。也就是说当恒流源用于电路偏置的时候，相当于在电路和干扰源之间连接了一个阻值为无穷大的电阻，干扰信号自然就被衰减很多了。而恒流源在保证交流阻抗无穷大的同时，又能保证恒定的电流输出，这一点它和普通的电阻又有不同。

- 如果 IC 内部的偏置采用恒压源，那也不错。因为恒压源可以保证参考

信号的稳定和纯净，电路自然就稳定。

- 三极管做的分立器件电路，它们的偏置电路一般都是通过一个或者2个电阻直接连接电源，假如偏置电阻的值是100 kΩ，那它对电源纹波的阻抗也就是100 kΩ，远远不能达到恒流源的“交流阻抗无穷大”。

不过，虽然用IC的纹波抑制比比较好，但是我在实践中也发现某些IC经常会出现由IC电源端进入的噪声。这种问题的处理方法类似于图11.7，为它的电源端增加一个*RC*滤波回路就会好很多。



原理说了一大堆，总之一句话：如果不是成本控制的很严格的话，尽量避免使用三极管、场效应管等分立器件设计线性电路，使用IC来设计电路，不仅电路简单，抗电源纹波、一致性也会很好。

最后再说说“通过电路的参考电压或者偏置电压窜入信号回路”这一条。这一条通常不是由于器件本身的特性引起的，而是由于电路本身设计不良造成的，是一种元器件搭配组合错误而导致的电源纹波干扰。这种问题只会发生在还没有完成产品化的设计阶段，而你不可能在市面上发售的产品中找到这种问题的电路。因为这种错误产生的影响太大了，以至于任何一个工程师都不会把有这种问题的电路，拿去批量生产。

图11.10是我曾经看到的一个真实的工程实例，这个电路在功率放大后的信号声音，总是感觉不自然，同时这个电路的杂音也非常严重。示波器发现图中运算放大器的直流工作点经常会飘来飘去，当飘得太多的时候，经常就出现切顶

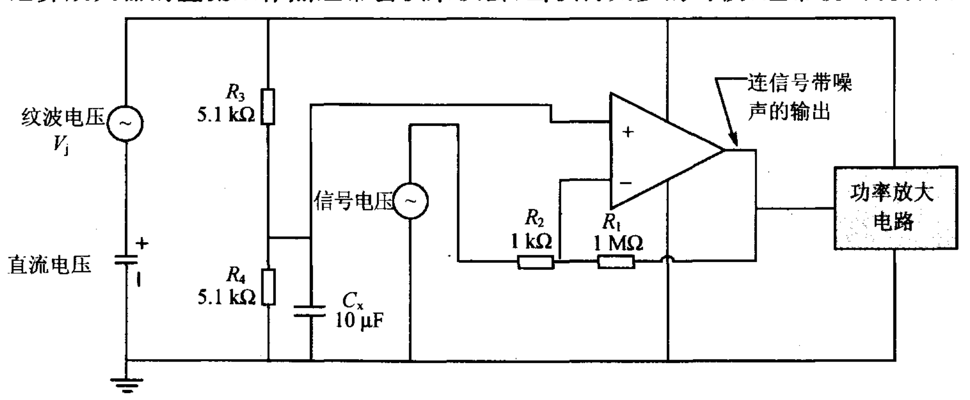


图 11.10

或者削底失真。一开始想了很久也不知道是什么原因。仔细看看电路图,发现为了降低成本,运算放大器使用了一个分压电阻组成的参考电源,取 $V_{cc}/2$ 作为运算放大器的参考源。并且还连接了一个 C_x 组成 RC 滤波电路。在静态的时候,用示波器测量运算放大器的"+"输入端,纹波不大,也就是 1 mV,因为在电源上 1 mV 的纹波算是很小了,感觉这个很正常,就没有去处理。但是这个放大器听上去就是不舒服。示波器观察运算放大器的输出有严重的低频干扰。

经过多天的测量和测试,终于发现,在音频输出重低音(音频输出重低音的时候,最能考验电源的负载能力)的时候,整个系统的电源都从 24 V 掉落到 23 V 左右,之后又可以回升到 24 V,所以运算放大器的参考电压也会有 0.5 V 的掉落。这 0.5 V 的变化,经过运算放大器一定增益的放大后,就会是很严重的大幅度电压变化。进而造成整个电路的直流工作点不稳定。

这样的电压掉落和回升可以看作是频率很低(大约 0.5 Hz~2 Hz)的纹波,这样的频率一般的 RC 滤波器根本没用。

改进措施如图 11.11 所示,加入了一个高稳定度的电压参考源 TL431,这个芯片是大量使用的芯片,也很便宜。加入这个芯片后,无论什么情况,运算放大器的参考电压都牢牢锁定在 11 V。立马解决了以上问题。这样改进后,电源依然会有纹波,音频输出重低音的时候,电源电压依然会掉落,但运算放大器的直流工作点就是纹丝不动,电路比原先稳定了许多。以上的例子,由于干扰来自 IC 外面,而且它影响到了 IC 的参考系。所以,IC 内部的差分和恒流电路对于干

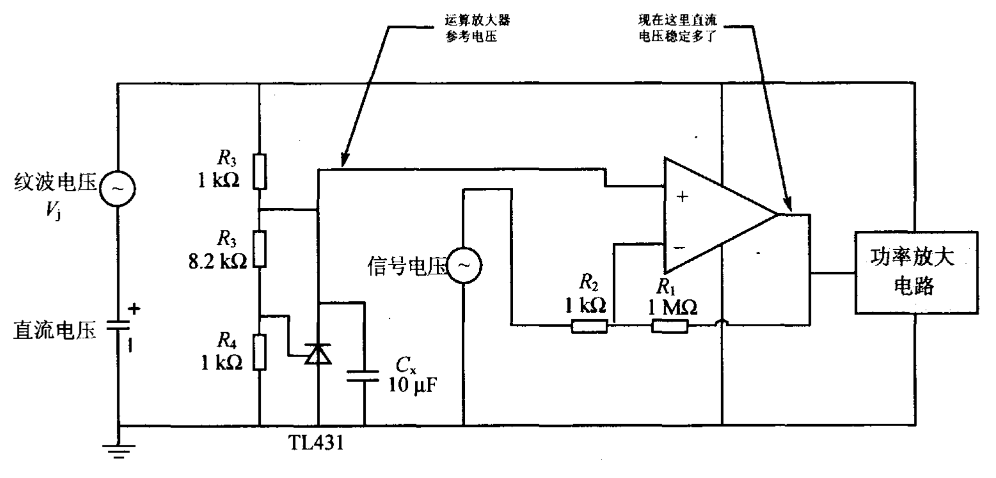
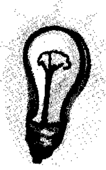


图 11.11

扰信号没有丝毫作用，解决 IC 外面的不稳定因素才是主要的。

如图 11.5 所示的问题，归根结底，其实也是这个问题。因为基极的电压不稳定，放大后，杂波非常严重。而图 11.6 加入了一个 *RC* 滤波器，滤除了偏置电流中的纹波，然后 2 个三极管都使用这个干净的偏置电压。所以其输出也会相对干净。



提示：

这 2 个例子，其实都变相承认了电源的电压波动是正常的，没有采用头痛医头，脚痛医脚的方法来解决问题。而是改进信号电路，通过增强其对干扰的适应性来解决问题。

11.6 搞定大功率电源

这里探讨的大功率电源，是指安装在电路板上，电流比较大的电源。其电流一般大过 400 mA 的电源。而大部分电路板上的 LDO 和 DC/DC，其电流都在 200 mA 以下。这样的小电流电源，设计非常简单，基本上在电路板上把线布通、元件没有安装错误就可以。当然，你可能说，400 mA 的电源也不算大呀。但是实际上，电路板上的电源，比较少有超过 3 A 的(电脑主机板除外)。而且电流增大以后，就不会像小电源这么简单了。因为电流增大后，会伴随很多问题。都解决后，电源才可以说是稳定的。

11.6.1 DC/DC or 线性电源?

经常看见很多人的设计，尤其是一些非民用设备的电路板，上面经常有一个很大的散热片，散热片上用螺丝拧着一个 7805。图 11.12 和图 11.13 是我见过的一个电路板的照片。该设备输入电压为 18 V。从散热片的大小来看，这个电路的电流为 300～400 mA。但是却配备了 6 800 μF 的巨无霸电容。我询问了一下电解电容的价格，一般 1 元钱可以买到 1 000 μF/25 V 左右的容量，那这一个电容就要 6 元，2 个就是 12 元。加上 78051 元、散热片 0.4 元、电路板至少 20 cm^2，要 0.8 元。这里一共需要 14.2 元。而就价格而论，我要达到 5 V/300～400 mA 的输出，如果用 DC/DC 芯片来实现的话，可以把价格控制在 6 元之内，而且电路板体积可以大幅度压缩。

当然就稳定性而言，这样的设计也没有什么不可以，事实上这个电路板工作

也很稳定。这样设计好处就是：

- 外行人，看了会觉得货真价实。
- 设计比较简单。

缺点是：

- 体积大。
- 价格贵。
- 电源效率低。



图 11.12



图 11.13

像这个设备，售价高。它有足够的价格空间来配备这些成本较高的东西。而且机器内部也有足够的空间来放置这些东西。但是现在的很多产品都是小尺寸设备，而且电力供应有限，必须提高电源效率。虽然用 DC/DC 芯片来实现这样的电源且达到相当的稳定度，其设计难度要比用线性电源高很多，但是仍然有必要探讨一下 DC/DC 设计的注意事项。

11.6.2 热 阻

既然是大功率电源，发热是避免不了的。线性电源发热大点，一般加散热器也能对付过去。用 DC/DC 芯片来做电源，虽然效率高了很多，可是这不等于它不会发热。而且用 DC/DC 芯片来做电源的设备，通常体积小，散热更加困难。所以，DC/DC 的散热难度有时候反而会超过线性电源，尤其是输出电流比较大的时候。

在工程学上评估一个东西散热速度的指标，叫做"热阻"。实际上，在专业实验室中，"热阻"也是最常用来评价散热性能的参量值。

那，什么是热阻呢?

所谓"热阻"(thermal resistance)，是指反映阻止热量传递能力的综合参量。热阻的概念与电阻非常类似，单位也与之相仿——℃/W，即物体持续传热功率为 1 W 时，导热路径两端的温差。对散热片而言，导热路径的两端分别是发热物体(IC)与环境空气。对 DC/DC 的 IC 而言，散热体是 IC 的封装材料，导热路径的两端分别是 IC 内部发热的硅片与环境空气和电路板。

实际上，为了更好地理解热阻的含义，可以把"热阻"理解为电路中的电阻，两个温度的差值相当于电压，而发热体的功率相当于电池的总能量。热阻低就相当于电池两端的负载电阻小，可以产生更高的电流，进而快速泄放电池的能量。所以在一定的环境温度和发热功率下，热阻小会让发热体表面温度更低，其性能也更好。

图 11.14 和图 11.15 分别是常用的 DC/DC 电源 IC MC34063 和 AP1501 的散热特性说明。从这两个规格书可以看出：

- 尺寸越大的封装，其热阻可能越低。这个很好理解，散热面积大了嘛。
- SOIC 封装的 MC34063，散热最差。虽然其规格书号称可以达到 1.5 A 的输出电流。但是从工程应用的实际来看，如果不加散热措施的话，超过 400 mA，芯片就已经很烫了。超过 600 mA 就可能引起热保护。因为这种封装，除了 IC 的出线是金属的以外，剩下的封装材料都是塑料





的。而塑料的导热性能肯定好不过金属材料。

- DIP 的热阻虽然好过 SOIC，但是和 TO－220、TO－263 比起来，还是差一大截。这仅仅是因为 DIP 的个头要大过 SOIC 好几倍，散热面积大。
- TO－220 和 TO－263 的 IC 底下都是一个金属片，发热的半导体材料就被紧紧地贴在金属片上，然后才打上塑料封装（见图 11.16）。所以其热阻低，低得很合理。

MAXIMUM RATINGS

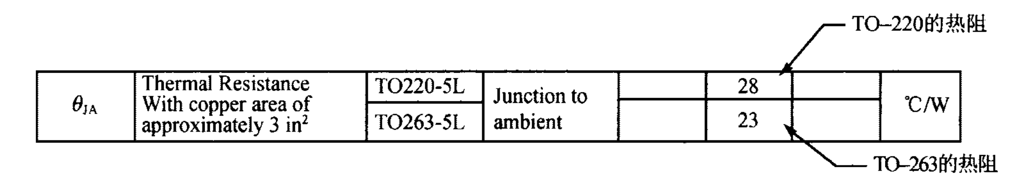
Power Dissipation and Thermal Characteristics			
Plastic Package,P,P1 Suffix			
T_A=25 ℃	P_D	1.25	W
Thermal Resistance	$R_{\theta JA}$	100	℃/W
SOIC Package,D Suffix			
T_A=25 ℃	P_D	625	W
Thermal Resistance	$R_{\theta JA}$	160	℃/W
Operating Junction Temperature	T_J	+150	℃
Operating Ambient Temperature Range	T_A		℃
MC34063A		0 to +70	
MC33063AV		-40 to +125	
MC33063A		-40 to +85	
Storage Temperature Range	T_{stg}	-65 to +150	℃

DIP封装的热阻

SOIC封装的热阻

IC的最高极限工作温度

图 11.14　MC34063 的散热特性和温度特性



θ_{JA}	Thermal Resistance With copper area of approximately 3 in²	TO220-5L	Junction to ambient		28		℃/W
		TO263-5L			23		

图 11.15　AP1501 的散热特性

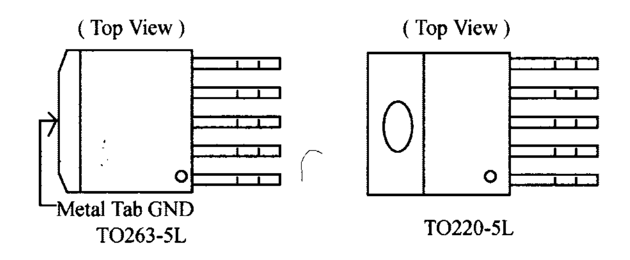


图 11.16

所以，超过 400 mA 的 DC/DC 电路，建议使用像 AP1501 那样的低热阻的 IC，这样可以利用电路板上的铜箔自然散热，不需要另外的冷却风扇或者散热片。TO－220 和 TO－263 那样的 IC，只要将其散热 IC 底座和电路板焊接良好，一般不需要额外的散热片。没有散热片意味着可靠性提高、节约人力加工成本。

11.6.3 DC/DC 工作频率

一般而言，DC/DC 芯片的输出电流越高，其工作频率也就会高一些。这是因为以下 2 点原因：

1. DC/DC 芯片的开关频率越高，其产生的电源纹波越小，同时也就越容易控制住纹波。

2. DC/DC 芯片外围一般都要配备一个储存能量的电感。这个电感的尺寸取决于 2 个方面：

- DC/DC 芯片的工作频率，工作频率越高的话，尺寸就可以越小。
- 电感都是用铜线绕制的，是有直流电阻的。所以电流越大，发热越高。DC/DC 芯片的工作频率越高的话，就越有可能使用小电感量的电感。而电感量小就意味着绕制电感的直流电阻小。

11.6.4 纹　波

上面说了，DC/DC 芯片的开关频率越高，其产生的电源纹波越小。可是这句话的意思并不意味着频率提高以后，纹波就很好处理了。实际上，DC/DC 的纹波处理掉的难度要远高过线性电源。

放置电容，这个在小电流的情况下，放上这些电容后，基本上就不会出现问题了。可是在大电流 DC/DC 电路里面，如果没有把握门道，即使放上一堆退藕电容，纹波也不会降下来。

如图 11.17，这是一个大电流的 3 A 3.3 V DC/DC 电路。这样的电路，只要元器件没有错，按照图 11.17 随便接起来都能跑。但是要想将电源纹波控制到一定水平的话就不那么简单了。因为无论什么 DC/DC 电源 IC，都是用开关管反复导通和关断来调整电压的，在开关管导通和关断的瞬间，由于电流瞬间的大幅度变化，很容易产生噪声(这种噪声用高速示波器很容易看见)。

我处理这种问题的经验是：对这种噪声最好的控制办法就是像图 11.18 那样，将几个元件的 GND 都以最短路径连接在一起。因为所有使用的导体都

不是超导体，都有一定的电阻，瞬间的大电流很容易产生压降而推高纹波。同时这些导体还有分布电感，在很陡峭的上升沿和下降沿作用下，可以表现出很大的感抗，也同样会拉高纹波。而单点接地就可以同时消除以上 2 个不利因素。

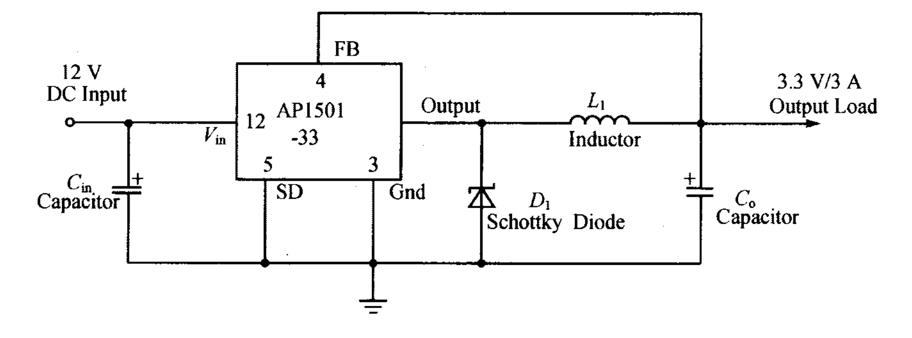


图 11.17

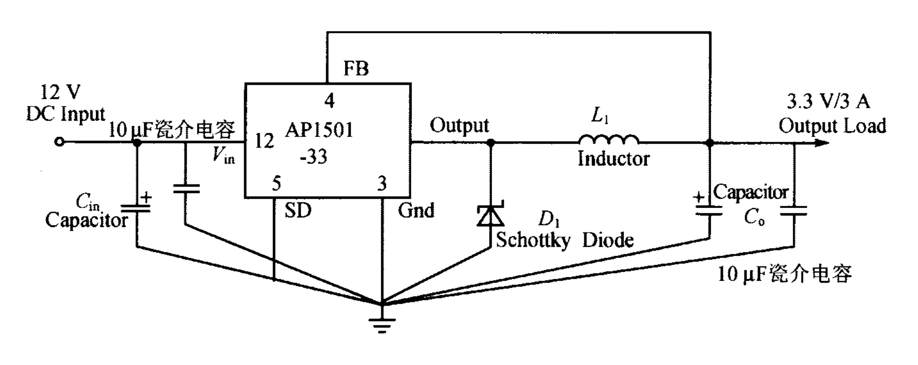


图 11.18

11.6.5 空间辐射

从工程实践的角度看，一般情况下，只要处理好了电源的纹波，DC/DC 电路产生其他干扰的概率也不大。但是在以下情况下仍然可能产生干扰：

- 距离 DC/DC 电路比较近的地方，有模拟电路的话，容易受 DC/DC 电路的干扰。
- 虽然某些电路距离 DC/DC 的距离不算很近。但是电路板上的易于受干扰的器件在电感磁力线的有效辐射范围内，也会产生干扰。这种现象在 2 片或者多片电路板层叠安装的时候，出现得比较多。

这里的 2 种干扰都是元器件的空间辐射干扰。它和纹波干扰一样，在电路上体现出来的干扰噪声的频率通常是 DC/DC 电路工作频率，或者是 DC/DC 电路工作频率的倍数。(因为 DC/DC 是在开关状态下工作的，其输出本身就包含了大量的谐波)。

处理办法如下：

- 惹不起，躲得起。重新设计结构，让二者离远一些就可以了。这虽然麻烦一些，却是根本的治疗办法。
- 如果因为机壳空间的限制，惹不起还躲不起。那就只好用隔离手段了。使用铁皮屏蔽罩之类的东西将干扰源封闭起来。更简单一些的，用导电胶布覆盖产生干扰的 DC/DC 电路，并用焊线将导电胶布接地。虽没有屏蔽罩处理得彻底，但却是最快捷的处理办法。

11.7 电源线上对外的辐射噪声

许多家用的电子设备，包括便携式设备，基本上都是通过 AC 适配器来供电的。AC 适配器与便携式电子设备之间的连接通过一根 DC 导线(如图 11.19 所示)。同样的，其作用类似于一根天线，它同其他的外界噪声源一样，发射噪声就来自这根导线。因此，此电源线的噪声免疫性测试就显得十分必要。

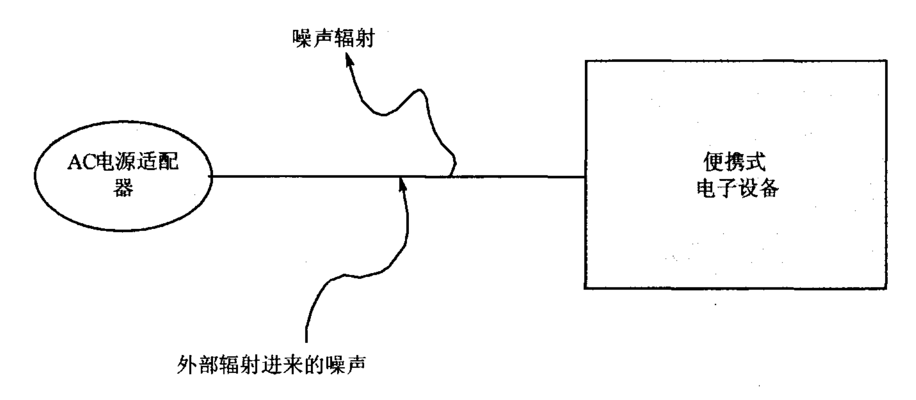


图 11.19

通常当对设备的噪声采用措施进行抑制之后，其免疫性会得到改善，因此研究噪声的产生十分重要。比如，从 DC/DC 转换器电源电路模块可以产生噪声，从内部数字处理电路模块也可以产生噪声。

在便携式设备中,DC/DC 模块和 DC/DC 转换器 IC 是电源电路不可缺少的部分。DC/DC 转换器所需电源通过开关模块控制产生,从而这种开关单元就变成了一种噪声源,这种噪声可以泄漏至直流电源线,且通过 AC 适配器导线发射出去。

同时,随着便携式设备功能的增多,所需半导体 IC 的种类也随之增多,比如像具有更高处理性能和大内存的微处理器。同时,随着处理量的增加和数据处理速度的提高,电路要求有更高速率的 IC 处理芯片。当 IC 在动作时,频率范围从几千到几百兆的宽频噪声就随之产生,并通过 AC 适配器的导线发射出去。

一般说来,便携式电子设备内部的噪声是通过 DC 电源线以共模的方式输出的,像在 AC 适配器导线上,其噪声可以用共模噪声滤波器进行有效抑制。在许多情况下,带有 AC 适配器的便携式设备所产生的共模噪声,一般是通过在线上加一个铁氧体磁环来抑制(见图 11.20)。尽管这是一种有效的噪声抑制方式,但是也存在许多不足,比如,加上磁环之后导线会变得沉重和庞大,且外观不雅。但是却可以让电子设备更稳定地工作。所以很多设备的线缆上都有这样的磁环。

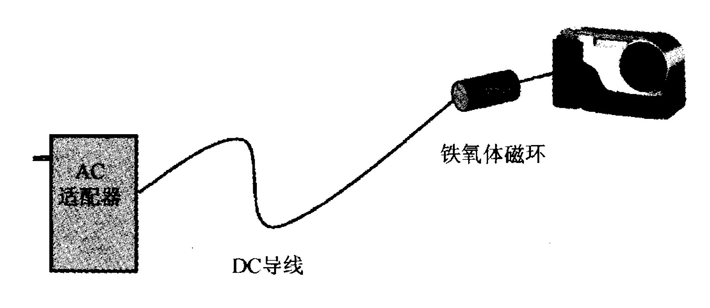


图 11.20

11.8 电源什么时候才算是正常的

正如本文在上面说的,没有纹波的电源不存在。有时候,可以使用镇压的办法,强制降低纹波;而有时候却要使用一些容忍的办法来解决纹波的问题。使用容忍电源噪声的方法,解决问题,前提是:电源的噪声是在可容忍的范围内。那什么样的噪声算是可容忍的呢?什么样的噪声算是不可容忍的呢?(见表11.1)



表 11.1

没有有源器件的纯负载电路。比如发热的电阻、电动机、电铃	这种电路，一般对纹波极度不敏感。纹波电压和直流电压的比值在30%以内，一般没事
大信号放大电路。输入的信号已经很强，整个电路的增益不大，比如小于 20 dB	这样的电路，纹波一般在 50 mV 之内都可以
小信号放大电路。输入的信号已经很微弱，整个电路的增益比较大，这样的电路增益一般大于 30 dB	• 小信号的前级尤其要注意纹波。要注意小信号的前级偏置电压(或者参考电压)的稳定性 • 小信号的前级电源纹波一般在 5～10 mV 之内都可以 • 偏置电压和参考电压最好能控制在 1 mV 以下 • 这种电路的电路板上，如果搭配了 DC/DC 变换器，而 DC/DC 的纹波控制得不好的话，经常会出现很难处理的杂音、图像条纹等毛病
大功率的放大、驱动电路或高端 CPU 等比较耗能的电路	• 要注意大功率负载可能造成电源跌落。而这种跌落千万不能影响到小信号回路的稳定性 • 这种电路最容易引发电源容量不足的毛病，在挑选电源供电方案的时候要小心 • 大功率的放大、驱动电路或高端 CPU 等电路，不仅自己耗能大，而且非常容易影响到小信号回路供电的电压稳定性和增大电源纹波

果粉工作室 Retro Workshop

12

从人体解剖理解 C++

我在阅读 C++书籍后，第一次从书籍上看见面向对象的编程思想。当时还犯嘀咕，干嘛要设计这样一个编译器，普通的 C 不是也很好吗？不过嘀咕归嘀咕，我还一向信奉“存在的就是合理的”。所以相信既然 Microsoft 和 Borland 等国际一流软件公司都有 C++的编译器卖(还有面向对象的其他语言的编译器，比如 JAVA、C# 和 Delphi)，自然也会有其合理之处。

用 C++做了几个工程。随着对 C++越来越熟练、对其理解也越来越深刻。同时也越来越觉得面向对象编程思想的伟大。我觉得 C++的很多概念和理解问题的方式，和自然界的规律是一致的。下面我就以人体解剖来做个比喻，诠释一下 C++吧！

12.1 Class 的可分解性和继承性

人体有很多器官，可以分成几种类别，比如消化器官，泌尿器官和血液循环器官等。为便于描述，统称它们为零件，我们的皮肤，骨骼也可以算是一种零件。每种零件功能各异，形状大小也大相径庭。但即使是这样，它们之间仍然有许多共同点。

每个人体零件都由一大堆细胞集合而成。而细胞要生存和新陈代谢，就必须有循环系统穿越其间。大的零件有大的循环管道，小的零件有小的循环管道。而很小的零件有微循环系统穿越其间。而每个零件的循环管道，总有一个血液的入口和出口。血液的入口所携带的血液总是含氧量高一些，出口的血液含氧量总要低一些，而代谢废物的含量相对高一些。

好了，有了这些共同点，就相当于 C++里面的基本类 CObject。人体的大部分器官都基于这个基本类而派生。当然也有个别不包含毛细血管的零件，比

如指甲，头发和单个的血细胞等。这个时候可以把它们认为是不基于 CObject 的类。

有了这个人体的 CObject 类后，就可以派生出很多不同的人体器官，比如从基本的 CObject 类派生出管道类器官，这种器官的显著特点就是一个入口和一个出口。入口和出口直接相通。比如肠子，胃，气管和大动脉血管等，也可以派生出非管道器官，比如肝脏和肾脏。这些器官的特点都是有一个进去的大血管和一个出去的大血管。还有一些其他的对外接口，比如肾脏还有一个输出端口——输尿管。

基于这个管道类，还可以进一步派生出很多零件，比如肠子，胃，尿道和大血管等。比如胃，在管道类的基础上加上几个 private 的函数，比如破碎食物、分泌胃酸、储存食物和自发蠕动等功能。而这些 private 函数对我们而言并不重要，我们只要根据胃的其中一个 public 输出，通知大脑的饱感和饥饿感，输入食物就可以。而破碎食物、分泌胃酸、储存食物和自发蠕动等功能都被"胃"这个 Class 封装了。这就是 Class 的封装性。

当然还有一些 Class 不是管道类的，而是像肝脏那样实心的，或者像是肺那样满是泡泡的器官，同样也可以理解为是从 CObject 类派生而来。

而一系列的零件可以构成人体的一个系统，可以把它看作是一个更大的 Class。比如消化系统的 Class、呼吸系统的 Class 和神经系统的 Class。而最后这一系列的系统，最后派生出一个完整的人体 Class。而不同的人体 Class，在各自的大脑里面安装上不同的操作系统，就有了不同型号的人，有的人适合做政客、有的人适合做商人、有的人适合做士兵、有的人适合做将军……

在 C++系统中，就是这样将一个个的 Class，通过不断的继承和派生，产生越来越复杂的功能。最后生成一个特定功能的应用程序。如果将其逐个分解，其组织形式应该是树形结构。

比如买一个电视机，拆开包装箱，可以看见几个 Class：电视机、缓冲泡沫、遥控器、说明书和 RF 电缆等。而每个 Class 都有可能是可分解的，比如遥控器，分解它之后，可以看见几个更小的 Class：外壳、导电橡胶和电路板。电路板这个 Class 又可以进一步分解为：PCB 板基、电阻、电容等元器件、红外线发射管和遥控 IC 等。如果要进一步分解遥控 IC，还可以看见……。反正，无论是电视机那样的复杂系统，还是遥控器这样的简单系统，都是由一个个微小的 Class，像

前面所说的人体那样，一步一步不断的继承和派生得到的。

如果用 C++来写程序，也是这样一层一层地构建 Class。比如图 12.1 这个 Microsoft WORD 中常见的查找替换窗口，依照我的理解，如果用 Visual C++来写，其 Class 构建层次顺序应该是像图 12.2 那样的：

查找和替换

查找(D) 替换(P) 定位(G)

查找内容(N)：

选项： 区分全/半角

替换为(I)：

高级 ▼ (M) 替换(R) 全部替换(A) 查找下一处(F) 取消

图 12.1

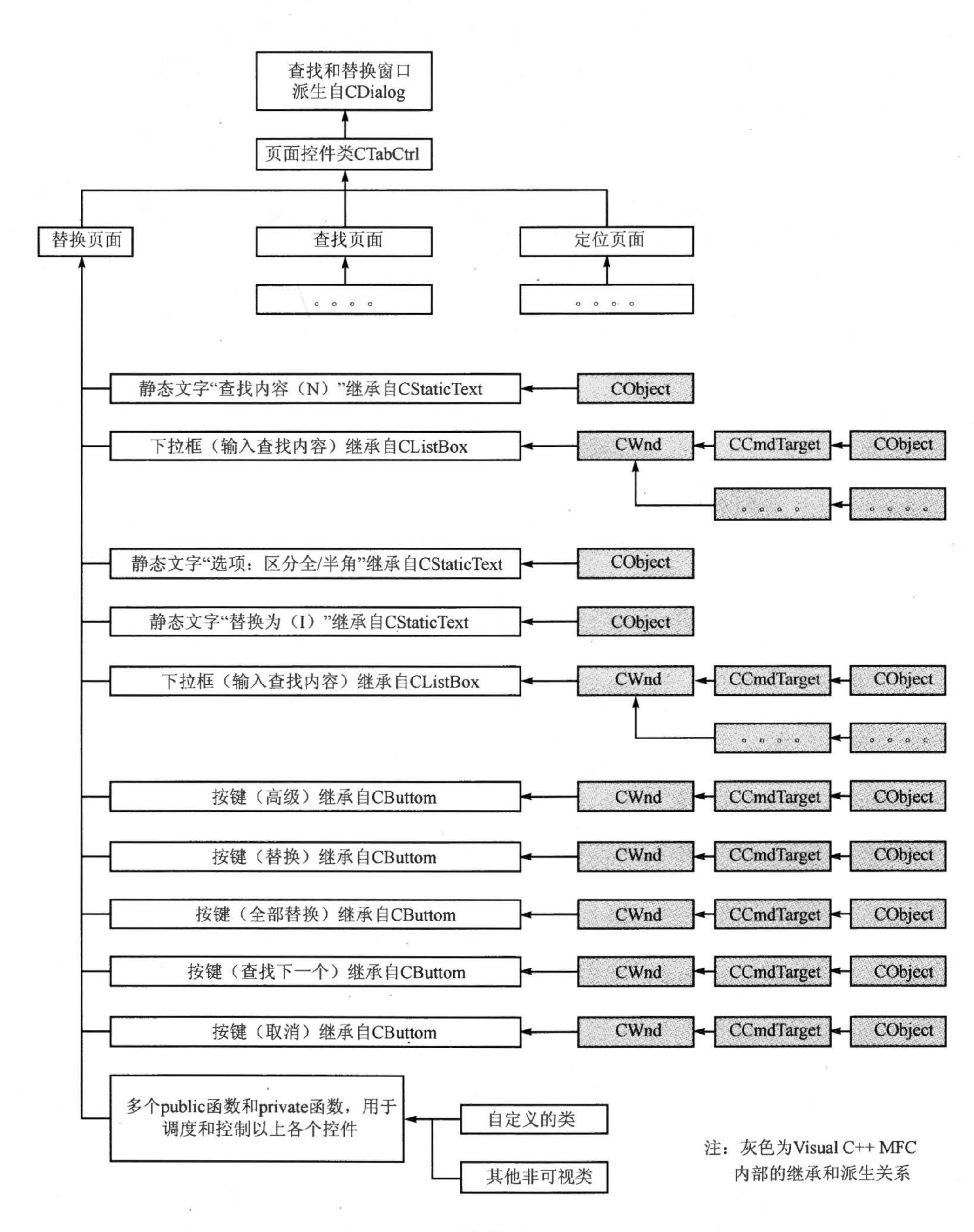
如图 12.2 所示，这个对话框内部的继承和派生关系，就像一棵树。CObject 都在这个树的叶子节点上。而图 12.2 这个对话框构成的 C++ Class，同样也是整个 WORD 程序的叶子节点。整个 WORD 应用程序就是由成千上万个这样的类似的 Class 子树(当然，绝大多数子树是非可视的类)构成的庞大应用程序。

12.2 接口的精炼性和 Class 的封装性

人体每个器官的接口都可以看作是 C++里面对外的 public 函数和属性。而人体每个器官的对外接口都很简单：

- 心脏只有 4 个血管作为对外接口。
- 肺只有 2 根血管和一个气管作为对外接口。
- 肾脏只有 2 根血管和一个输尿管作为对外接口。
- ……

而各个器官通过这样简单的连接，将各个部件连接在一起，构成一个完整的人体。各个器官在这些简单的接口后面，都包含了内部完整的工作流程和内部构件。这些流程和构件可以看作是私有的，而那些用于连接其他器官的管道，可



查找和替换窗口
派生自CDialog
页面控件类CTabCtrl
替换页面
查找页面
定位页面
。。。。
。。。。
静态文字"查找内容（N）"继承自CStaticText
COject
下拉框（输入查找内容）继承自CListBox
CWnd
CCmdTarget
CObject
。。。。
。。。。
静态文字"选项：区分全/半角"继承自CStaticText
CObject
静态文字"替换为（I）"继承自CStaticText
CObject
下拉框（输入查找内容）继承自CListBox
CWnd
CCmdTarget
CObject
。。。。
。。。。
按键（高级）继承自CButtom
按键（替换）继承自CButtom
按键（全部替换）继承自CButtom
按键（查找下一个）继承自CButtom
按键（取消）继承自CButtom
多个public函数和private函数，用于调度和控制以上各个控件
自定义的类
其他非可视类
注：灰色为Visual C++ MFC内部的继承和派生关系

图 12.2

以看作是一个Class中Public类型的变量。这些私有的流程和构件可以大大简化外部管理这些器官的难度。因为从这些器官外面看来，只需要在其输入接口输入特定的东西，然后在其输出接口就能得到想要的结果。这就像一个被封装过的IC一样使用简单。

同样的，《软件工程》里面一再要求，函数的对外接口要尽量简单。只有简单的接口，才能做到程序的可读性好、健壮性和易于使用和维护。

拿一个超薄式的数码相机在手上，你就会发现，操作实在简单。什么都帮你搞好了，你要做的，只是打开电源，按下快门而已。数码相机的设计者就是要将用户的使用难度降到最低，所以它将很多的感光、对焦和文件存储等功能都全部封装进去了。如果用自动档的话，那能用到的接口就只有电源按键和快门按键。

数码相机为了做到傻瓜化，必须自适应很多东西，要能够自动适应光线的强弱变化而自动调节光圈和闪光灯。当光线很强的时候，不开启闪光灯，同时光圈缩小，快门也变小。当光线弱到一定程度的时候，曝光时间自动变长，如果曝光还是太弱，则光圈会自动开到最大，甚至会自动开启闪光灯。这样，照相机就可以适应很宽范围的照度了。它还要能够适应被拍摄者和照相机的不同距离而自动调节焦距。CPU和软件根据测得的距离，自动决定镜头中几个镜片的相对位置而达到调焦的效果。而这一切都是作为数码相机内部封装的私有函数，在照相机内部自动处理完成。所以这样的照相机，才能够保持接口的简洁，做到傻瓜化。

佩服大自然的伟大，人体的器官也是这样设计的：

- 消化系统的Class，输入的是美食，输出的是不能利用的渣滓。有用的成分，都送到了循环系统。中间的过程，可以不管。
- 流入肺部的血液是含氧量低的血液，流出的是含氧量高的血液。中间的过程，可以不管。
- 肾脏也一样，含有毒素的血液流入，出来就是比较干净的血液。而滤出的毒素就通过输尿管排泄掉了。中间的过程，可以不管。
- 循环系统也有这样的封装性：当剧烈运动的时候，会自动调大心脏的输出功率（脉搏加快），同时还会通知呼吸系统自动加快呼吸的节律，以氧化更多的红细胞。而人体平静下来后，心脏也会自动减小输出功率，呼吸也趋于平静。这中间的过程，可以不管。

12.3 构建函数和析构函数

生物体和C++一样，也有构建和析构，你相信吗？

人一生下来，主要就是在长身体。他(她)需要吸收来自大自然(至少也是间接来自大自然)的能量和营养物质，各个器官朝向成熟的方向成长。这个可以理解为是一种构建。这个类似于软件构建一个Class，需要占用CPU、消耗内存和打开文件等资源。

而人体死亡后，通过人工的方式或者通过天然的方式，若干时间后，人体的构成元素会化整为零，全部化解为很容易被大自然重复利用的低分子量的有机物或者无机物。逐渐回到自然界，重新加入整个自然界的物质循环过程。若干时间后，可能一部分原来某人身体上的元素，被放置在了天空中的小鸟身上或者汇入涓涓的溪流，灌溉地上的小草……

而在C++中，一个Class的构建，首先需要分配一段内存。然后可能要打开一些资源，比如外设或者硬盘文件等，之后才可以正常工作。而Class的析构，则是和以上相反的一个过程，首先按照Class的继承层次，从叶子节点开始，逐个销毁子类。把每个Class子类占用的资源归还给应用程序或者操作系统。最后主Class也销毁，占用的资源同样也归还给应用程序或者操作系统。当应用程序的最后一个Class销毁的时候，就意味着整个进程被销毁。这个进程占用的所有资源也被操作系统回收。操作系统得到这些资源后，又可以启动下一个相同的或者不同的进程……

12.4 Class的健壮性

所谓健壮性，简单的说就是稳定性。在一个Class实现之后，不仅要完成其主要功能，还要能抵御各种异常条件和异常输入。

一个封闭的Class，在其精炼的接口被定义好之后，就比较容易做好健壮性。因为精炼的接口背后的含义是数据处理权限的准确划分。哪些数据要在哪个接口里面处理，很容易理解。这样即使数据处理出了差错，也很容易找到是哪个模块出了问题。这个就好像在工作中，各个员工的职责分明，每个人的工作就不容易出差错。而如果责任不明确，出问题的时候就会很麻烦。

一个良好的程序，除了需要能处理正常范围的输入数据以外，还能够处理一些比较离谱数据，至少要对这些离谱的数据给出错误的提示。比如一个开根程序，正常的输入范围是大于 0 的数据，可是输入一个负 1 的话，按道理这个时候开根无法运算，但是此时程序不能崩溃，而应该出错退出或者给出错误提示。

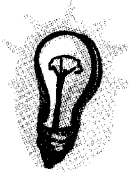
以上面提及的数码相机为例，为了保证产品的稳定性，数码相机的设计者通常还会关心：

- 外壳是否坚硬，是否抗冲击？
- 电池的放电过程有没有保护，不能够过放电，所以电池耗尽要自动关机。
- 存储卡和电缆插拔引入的 ESD 干扰脉冲是不是会损坏电路板？
- 电池是否有足够的续航时间？
- ……

这些相当于设计 C++程序的时候，所关心的“健壮性”。

同样的，人体也有很多体现健壮性的设计：

- 当循环系统输出功率已经最大，心脏已经超负荷的时候，会通过神经系统让人感知疲倦。如果人体在疲倦下仍然不停机，则会暂时强制人体进入休眠模式（昏厥），以保护人体的零件不超载。
- 肌肉不停的运动，会产生酸胀的感觉。促使人体减小运动的幅度或者停下来。
- 吃了带毒性的东西，会产生呕吐或者腹泻，以尽可能排出毒素。
- 人受伤后会产生疼痛的感觉，某些身体部位炎症也同样会产生疼痛的感觉。以便让人感知危险的存在并远离危险。





提示：

面向对象编程思想的伟大，就是因为它还原了大自然界中事物的组织方式和层次结构。这样的编程方式，使得程序更易于管理、更有序和更稳定。

13

电子产品设计阶段的成本控制

设计电子产品的过程一般都会对所设计的产品进行成本控制，总是尽量简化电路，减少元器件以及尽量简化安装工序。一般这些简化都是针对最终产品的。但是作为开发工作本身也同样需要成本，也需要进行成本控制。尤其是一些产量小、附加值高的非消费类产品，其大部分成本就体现在研发阶段。

电子产品的开发成本一般包括以下几部分：

1. 项目可行性分析费用。
2. 联系元器件厂商，获取器件资料费用。
3. 元器件费用。
4. 开发人员薪资。
5. 样机测试费用。
6. 时间成本。

以上几点的成本控制和管理在很多工程管理的书籍和资料中都有详细介绍，本章只是根据笔者所做过的电子设计工程补充的一些看法。

进入项目可行性分析阶段后，项目管理人员应该仔细做好项目规划工作，一个项目的成功与否，一般取决于该项目的技术复杂性和成本复杂性。为了避免由于不可预知的工程复杂性而导致的项目流产，项目管理人员在制定设计方案的时候需要招集各方面的人员，把该项目仔细地分解开来，然后针对这些子项目逐一探讨分析，仔细地权衡各方面因素，看看是否可行，成功的代价如何，只有每个子项目都做到有把握后才能将整个项目推入实施阶段。

在项目通过可行性论证后，就转入正式的开发阶段。项目管理人员需要制定详细的开发技术规划，一个项目的设计思路确定之后，该产品的开发成本、制造成本和维护成本也就大致确定下来了。所以一个不良的设计规划，犹如军队在出兵之前没有一个良好的作战计划一样，其结果往往是灾难性的。这个阶段项目负责人要和合作的开发人员充分交换意见，根据开发人员的数量和专长将

项目分解开来，让每一个工程技术人员完成本项目的一部分工作。

项目开始运转后一定要做好全套设计文档。文档中要明确每个开发者所必须完成的功能和相互之间的接口。同时也要要求每个开发人员为自己所开发的模块做好技术文档。表面上看这个工作增加了开发工作量，但这样不仅有利于该项目今后的扩充维护，同样也有利于该项目的测试工作。事实上降低了项目在这些方面的成本，项目规模越大，这方面投入所体现的效益就越明显。同时这项工作还可以降低由于开发人员流动带来的工程扩充维护的风险，因为技术文档越多、越详细，继承该工作的技术人员所付出的时间代价就越小。

在元器件选择方面，应尽量使用标准器件或易于采购的器件。因为这些元件产量大，价格好，供货渠道也多，对于降低硬件成本有显而易见的好处。尤其在所设计的产品产量不会很大的时候更应该如此。在设计一些高附加值，小批量产品的时候，尽量使用市面发售的硬件模块和软件模块来设计，这虽然加大了一些投入，但总的来说，压缩开发时间，让产品更快面对市场带来的效益会大于这些投入。同时模块化设计还可以提高产品的设计质量，更可以将开发人员的精力集中于高层次的设计上，提高他们的成就感。

尽量使用各种 EDA(电子设计自动化)工具。综合使用各种 EDA 工具来完成设计，可以大幅度加快开发进度，减少差错，提高工程质量。一提起 EDA 工具，很多人就会想起 Protel、Orcad 和 PowerPCB 等电路板布线软件。其实这些工具不仅包含原理图和电路板布线，一般也包含了可编程逻辑器件(CPLD、FPGA)设计、信号仿真等模块、自动电器检查和布线完整性检查等。充分利用这些功能往往可以在设计早期阶段就发现很多构思和图纸上的缺陷。对减少设计阶段的返工和修改有事半功倍的效果。此外，还有一些其他种类的软件，虽然不是专门的电子设计软件，但依然可以借用。比如：如果项目中包含了一些复杂的数学算法(如模糊逻辑和人工神经元算法等)，我们还可以动用专门的数学 CAD 软件——MATLAB 先仿真一下信号处理流程，然后根据仿真的结果来设计相关的硬件和软件。这样就节约了很多在目标机上反复写片、反复调试算法的时间。对于一些不很复杂的数学问题，可以使用软件——Microsoft EXCEL 进行数据分析和数据绘图。如果能够灵活利用 EXCEL 内部形形色色的数据统计和分析函数，也能解决很多问题。

在设计电路的时候，修改硬件在所难免。为了便于电路修改，要注意电路的可塑性。电路的可塑性是指电路的可修改能力。如果电路便于修改，会减少很多开发人员更改电路的低级劳动。提高电路的可塑性一般有以下几种方法：

1. 如果是对成本不敏感，且用了比较多的CMOS和TTL电路，或者这部分电路有较大修改的可能，则要尽量使用CPLD(可编程逻辑器件)来实现，因为CPLD本身具有可修改的特性。用它们实现数字逻辑后，要改变逻辑关系的时候，只要在计算机上修改它们的逻辑描述文件，然后经过编译，写片就得到一片新的逻辑关系的集成电路。就像修改软件一样容易。用CPLD代替不同的普通数字集成电路后，有利于减少元器件种类，方便采购和库存管理。

2. 试制过程中适当在电路板上多留一些资源，比如CPLD的容量要比预期的容量稍大一些，单片机的ROM、RAM和I/O端口等资源都要留适当的空余。因为在设计过程中随时会有很多不可预见的情况发生，解决这些问题，通常会增加对硬件资源的需求量。如果没有在电路板上保留适当的冗余资源，将不得不在电路板外面再搭一块小电路板。事实证明，这样修改电路不仅容易给电路引入干扰信号，而且由于电路板经常翻来覆去的测试，会导致线缆折断，这会大幅度降低试制样机的可靠性，同时会使技术人员的很多时间花费 在反复修改电路的简单劳动上。

3. 如果在设计的时候，某些地方的某个元件不确定是不是要安装，我建议就把需要安装的元件和不确定是否需要安装的元件一起画在电路板上。不确定的元件在原理图上标示出来，这样电路板做好之后，你就可以进退自如了。现在的很多新产品都需要改2～3次电路板。这样做到话，等到第3次做电路板，一般就会知道什么东西可以留着，什么东西没用了，而如果没有在开始就这样可进可退的设计的话，万一要增加元件，电路板会改得很难看。电路板难看通常意味着工作的可靠性差。

编写单片机和嵌入式系统软件的时候，应该尽量使用高级语言来编写。现在许多开发单片机的技术人员依然保留10年前的开发方法。使用汇编语言，逐条指令编写，并且手工分配单片机中的内存资源。这种方法在写小规模软件还可以，可一旦软件规模变大之后，由于汇编语言的低可读性和低结构性往往会让设计人员被自己软件中的复杂逻辑关系搞得晕头转向，在一个软件的逻辑关系混乱后，要保证写出来的软件能安全运转是很困难的。

我们推荐使用C语言来写软件单片机代码。每个类型的单片机一般都有其配套的C语言编译器。如果你的CPU够猛，甚至可以用C++。用C/C++语言有以下几个好处：

1. 可以大幅度加快开发进度。

2. 可以实现软件的结构化编程，它使得软件的逻辑结构变得清晰、有条理。

3. 在写一些数学算法、循环和判断语句的时候有比汇编语言高得多的效率。
4. 省去了人工分配内存资源的工作，在汇编语言中我们不得不为每一个子程序来分配储存变量的空间，这是一个复杂、乏味又容易出差错的工作。而使用 C 语言，只要在代码中申明一下变量的类型，编译器就会自动分配寄存器和内存，低级重复易出差错的事情都由计算机代劳了，根本不要人工干预。
5. 因为循环、判断语句和变量名字都使用自然语言，因此写出来代码的可维护性和软件可读性很好。这样当需要更换软件开发人员的时候，可以比较容易地进行代码移交和消化。即使没有更换人员，他们在维护自己的代码的时候也比较容易。
6. 当写好了一个算法后，以后需要在不同种类的 CPU 上也需要这个算法时，可以直接引用原来的代码，再配合这种 CPU 专用的编译器重新编译就可以了。这样可以实现软件的低成本跨平台移植。而汇编语言在这种情况下，除了重新写代码，没有别的办法。因为 C 语言的这种特性，如果在设计复杂算法时（比如视频/音频信号处理、复杂的计算和逻辑），可以在 PC 机上使用 Visual C＋＋、C＋＋ Build 等 x86 平台的 C 语言编译器设计、调试该算法的代码，调试成功后就可以移植到单片机的软件系统中去。由于 PC 机上有充足的系统编程和调试资源，开发调试进度会大幅度的提高，这样就节约了软件的时间成本。
7. 在团体写作的软件中，软件接口容易做到规范统一。
8. 网络上有很多开源的软件模块，基本都是用 C 语言写的，直接利用它们，事半功倍。

虽然使用 C 语言写出来的代码会比汇编语言所占用的空间大 5%～20%，但是由于半导体技术的发展，芯片的容量和速度有了大幅度的提高。在这种情况下，代码占用的空间差异已经不是很关键了。相比之下，更应该注重软件是否可以长期稳定运行的能力，注重使用先进开发工具所带来的时间成本的优势。

现在 C 语言其实已经很普及了，使用 ASM 语言做设计的人，也在日渐减少。但是还有一种方法可以大幅加快设计的速度，那就是尽量利用好编译器附带的库函数。尤其是 ANSI 库函数，该库函数是经过国际标准组织备案的函数库，一般和编译器绑定在一起发行。这些函数的质量都很好，极少会有 bug。我曾经见过一个菜鸟，写了一堆 for 循环来移动字符串，却不知道 ANSI 函数库里

面有一个 strcpy( )函数。用很多行循环，来实现清除一段内存的数据，却不知道 ANSI 函数库里面还有一个 memset( )函数。而用这些函数，一行就可以替代他那一堆代码。

更宏观一些，还可以用一些开源的代码。比如实现一些常用的数据结构(比如队列、堆栈和树)，可以用开源的 STL。要处理常用的图片，网上也有大把的图片处理开源软件包。如果能用好这些东西，又可以在前面的基础上大大地再进一步。

避免重复设计。有一句话，叫做："要想造汽车，就不要去设计车轮；要想设计军舰，就不要去琢磨怎么炼钢"。这句话的意思，决不是说汽车就可以不要轮子、造船可以不要钢板。而是说设计的时候要分清主次，把设计资源投放到接近目标的区域中去。设计时所用到的低级资源，比如钢板、车轮、螺丝和弹簧等，都可以去外购。如果条件允许，直接对外采购汽车底盘和引擎也未尝不可。采购外部商品化的模块，可以让某个模块快速稳定下来，减少 bug 的含量。因为别人能拿来卖的，一般都是经过了一段时间的锤炼，性能比较稳定了。所以，外购带来的最大好处是，充分利用了社会化大分工，节约不必要的研发开支。

14 物以类聚,信号以群分

14.1 在设计之外的"物以类聚"

生活中,大量事物都服从正态分布。就是说,在一个群体中,绝大多数的个体具有接近总体均值的特征:大家有着基本一致的情况和爱好。

例如,绝大多数成年人每天睡 8 小时,不会是你睡 0 小时,他睡 1 小时,他睡 2 小时,他睡 12 小时,而是均匀地分布于 0 至 12 小时内。又如,绝大多数人白天活动,夜里睡觉,断不会是一半的人白天活动,夜里睡觉,一半的人白天睡觉,夜里活动。这可不完全是因为昼夜分明之故。假如没有黑夜,人们活动和睡觉的时间也会集中并固定于某个时段。这些都是为了避免外部性。换句话说,如果历史上真有个体不是接近总体均值这样的种群,如果历史上真有远离总体均值的个体,那么也早让外部性给淘汰了。

有句古谚:"物以类聚,人以群分。""物以类聚,人以群分"的现象实在不少见。

- 找对象结婚,要门当户对。
- 富人大都集中在富人区,穷人大都集中在穷人区。
- 世界有国家和民族地区之分,具有相同文化和价值观的人群集中居住在一起,居住在一起的人们形成相同的文化和价值观。
- 中国人到了美国,他们集中居住于唐人街。
- 医院里,集中的都是身体有问题的。而问题更严重的,多集中于重症监护室。
- 太阳系中的几个行星,可以分为类地行星和类木行星。类地行星就是一大堆的石头,以固体为主体组合起来的星球,比如:地球、水星和金星等。类木行星就是一大堆气体,以气体为主体组合起来的星球,比如:木星、土星、天王星和海王星。这种规律,在别的恒星系中也一样。

旧时北京有“东富西贵”的谚语，反映的是历史上北京的一种特殊区域文化现象，同时也反映出老北京内外城的商业经济和政治文化分布的基本势态。清末满族学者震钧，他在其所著《天咫偶闻》卷十记曰：“京师有谚云：‘东富西贵’，盖贵人多住西城，而仓库皆在东城。盖富人多喜居东城。”而现在的北京，中央机关大多数在西偏北；外企和商务大多集中在东边；知识分子大多集中在北偏西；后来亚运会之后，北边也成为重要发展地区；中等层次的商业人士喜欢往正北边去；南边以低等级的商业活动为主，如小商贩、外地到京发展的人士等。同样的，上海的高端商务区都集中在浦东，中端商务区多集中于虹桥、中山公园一带。深圳的电子行业也多扎堆于深圳的西半边，而扎堆更集中的地方，就是赛格附近。也正因为如此，我每次去深圳出差就极少去深圳的东半边。

我们知道，在太阳系中，太阳占据了太阳系中的绝大部分质量，而剩下几个行星的质量虽然有大有小，但是和太阳比起来都算小毛头。这样整个太阳系的主要引力场都由太阳提供，行星与行星之间的引力场干扰比较小，这些行星才可能年复一年稳定地绕太阳运转。也正因为如此，才有了地球的生命和文明。想象一下，如果在几个小的行星中间突然插入了一个质量很大的行星，比如木星的质量变成了 1/2 太阳的质量，那木星对其他行星的干扰就会变得非常强大，导致几个行星的轨道面目全非。因为这样的话，行星之间的引力场互相牵扯，行星的运行轨道会变得非常复杂。各个行星也就失去了稳定运行的基础条件。

我们都知道姚明的个头远远高过一般成年人的身高，所以注定了他和一般人群的兼容性比较差。他的衣服鞋子都需要特殊定制，一般的桌子对他来说都太矮、一般的椅子对他来说都太小。而这些对一般人都不是问题，因为一般人的身高体重和群体大众的平均值接近。所以他注定不可能从事一般人可以从事的职业，也正因为如此，所以他“类聚”到了篮球队，因为在篮球队里面，他的特点和大多数篮球运动员类似，不仅不会出现兼容性的问题，而且还可以配合得很好。

但是因为姚明是一个人，有思想的人，他可以指导自己向有利于他自己的地方流动，而我们要设计的机器不行，因为机器没有智慧，没有思想。你怎么排列组合它就怎么工作，这个工作可以是有条不紊的，也可以是一团乱麻的。为了让没有思想的机器能够正常工作，所以就需要用设计者的思想来安排机器的每一个细节，来对机器和设备内部的每一个元器件进行仔细的排列组合，使其组合在一起后可以相安无事，共同稳定地工作。

14.2 物以类聚，信号以群分

电路里面的信号也是如此，不同的信号有不同的阻抗、辐射强度和干扰耐受力等，所以和现实社会中人类一样，为了稳定与和谐，也需要“物以类聚，信号以群分”。

100多年前，磁带录音机算是比较流行的家电，为此我也经常修理录音机，积累了很多经验和感受。在修理的时候发现磁头的外壳总是有一个接地线；录音机的机芯有一个接地线；卷带机构的电动机也有一个接地线。如图14.1所示。

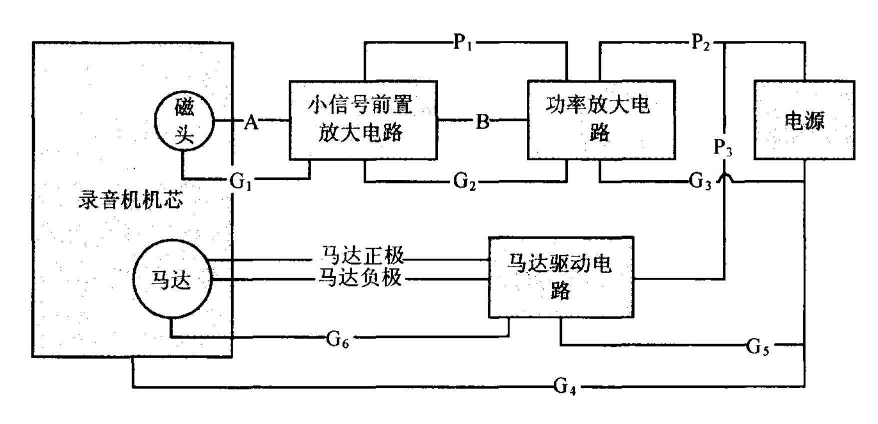


图 14.1

表 14.1

信　号	相关解释	
A	磁头出来的微弱信号	
B	经过初步放大的音频信号	
$P_1 \sim P_3$	DC 电源的正端	
G_1	这几个信号线都是系统内的接地线	这个信号用于将磁头的外壳接地，接地线拉到前置放大器的输入端。磁头虽然安装在录音机机芯上，但是它的金属外壳却不和机芯直接相通。磁头外壳接地是为了防止外部的电场和磁场干扰磁头，保证信号的纯净。因为磁头的输出信号本身非常微弱，一点点的干扰就会严重影响声音的品质

续表 14.1

信　号	相关解释	
G_2		前置放大器的 GND 再连接到功率放大器输入端的 GND
G_3		功率放大器输出端的 GND 再连接到电源的 GND
G_4		机芯接地线,它直接接到电源的 GND。因为机芯是很大一块金属,很容易感应各种干扰信号,比如马达的电火花辐射和天空的辐射等。把它接地就是为了尽量降低由这块金属引入的辐射干扰
G_5		马达驱动电路的 GND 再连接到电源的 GND。这个线上面有比较强的干扰
G_6		马达的外壳接地线,这个接地线可以最大限度地屏蔽马达内部电刷的火花干扰。这个线上面有比较强的干扰
马达正负极	因为录音机的马达大部分都是有电刷的马达,很容易产生电火花,所以这个线上面有非常强的干扰。也正因为如此,这 2 根线通常不会很长	

从上表 14.1 可以看出,同样是接地线,有的是微弱信号的接地,有的是强干扰的接地。接地的途径是:弱信号的接地→中等强度的信号接地→大信号的接地→电源。而且有用信号回路的接地和干扰信号回路的接地是严格分开的,最后到电源那里才连接在一起。

- 弱信号的接地和中等强度的信号接地连接在一起,是因为二者的信号强度基本类似,相互之间的容忍度基本类似。
- 中等强度的信号接地和大信号的接地连接在一起,也是因为二者的信号强度基本类似,相互之间的容忍度基本类似。
- 大信号的接地和电源地接在一起,还是因为二者的信号相互之间的容忍度基本类似。
- 马达电路的 GND 和磁头的 GND 之所以不能乱接,就是因为一个是很敏感容易被干扰的弱信号器件,一个是会时时辐射出强干扰的器件。所以二者放在一起,出问题的可能性极大。

这个原则,不仅录音机上是这样,很多同时包含微弱信号和电机的电路,都是这样设计的。比如 CD 光盘驱动器、DVD 机和硬盘等。

之所以这样设计,就是因为要把产生辐射的部分和容易被辐射干扰到的部分尽可能地分割开来。相同量级和功能的元件集中在一起;类似量级,对噪声容忍度差别不大的功能模块可以作为邻居;而那些喜欢干扰别人的模块,需要单独

设计在一组。这样电路就可以比较稳定地工作。而如果有谁敢挑战这个规则，把电路设计成类似图 14.2 或者图 14.3 那样的形式，那最后的结果一定会死得很难看。

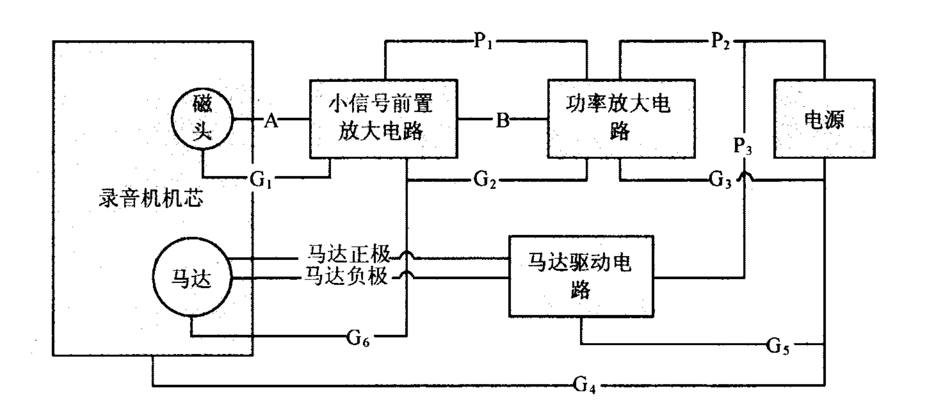


图 14.2

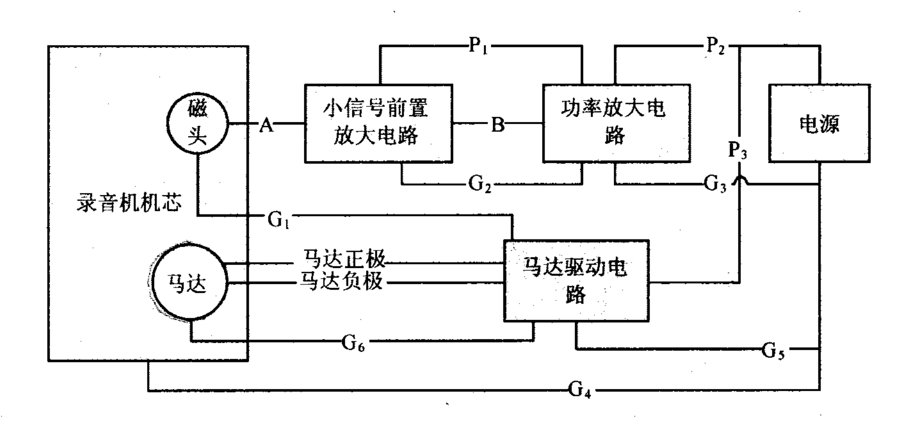


图 14.3

电路中这样的例子很多，比如 CRT 电视机中高压包的那根红色的高压输出线，从来都是单独走一根到显像管的高压嘴。从来不会和其他的线并排走，而且即使是单独走，也经常要用很大的尼龙支架套在上面，以保证这根红色的高压线和电视机中的任何其他部分电路至少有 25～45 mm 的距离(包括显像管除了高压嘴以外的任何地方)。因为它的电压实在太高了，和那些低电压的电路放在一起就一定会放电打火。把它单独走线是唯一的选择。

类似的,很多 32 bit CPU 的引脚也分为地址总线、数据总线和控制总线。这样区分不仅容易理解和交流,也因为这三种信号各有不同的速率和辐射强度等差异,所以在电路板布线的时候,一般都把一种总线单独汇成一排走线,这样走线不仅美观,也降低了电路的对外辐射,同时还便于统一处理外部辐射对这一排总线的干扰(如果有干扰的话)。

如果有留心电视机内部的话,还可以发现,电路板上的高频头和中频部分通常集中于电路的一侧,而电路板的另外一侧通常会有开关电源、行场扫描和高压包等部分。因为把高频头、中频和色解码等弱信号电路放在一起,它们之间是半斤八两,不容易互相干扰。而开关电源、行场扫描和高压包都是很容易产生对外辐射的部分,它们之间也是半斤八两,聚集在一起也不容易出问题。所以这样的安排似乎已经形成了电视机电路板的通用规则。这样安排同样是为了降低干扰。

图 14.4 是一个手机电路板的例子。图中的 A 是模拟基带处理部分,被一个铁皮单独屏蔽起来。B 是数字基带的 CPU 处理单元,是一个高速的 ARM 处理机,也被一个铁皮单独屏蔽起来。C 则是一个 GSM 的收发模块和 GSM 的协议栈芯片,因为它的辐射最强大、接收的信号也最微弱,所以被一个完整的铁皮包裹起来。其他没有用铁皮框起来的部分则是那些不容易干扰别人,也不容易被别人干扰的部分。

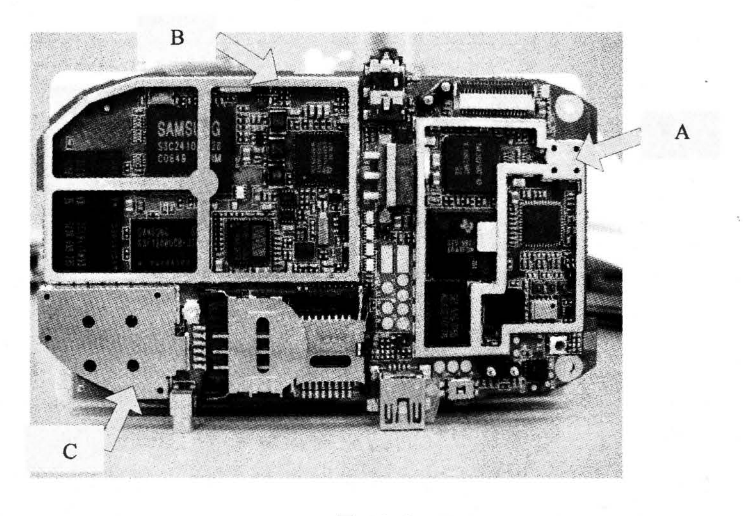
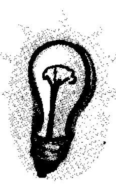


图 14.4







提示：

- “物以类聚，信号以群分”的主体思想就是要使个体具有接近整个小群体均值的特征。
- 各个相邻的群体也最好要有类似的特性，如果做不到这一点，就必须强制隔离。（图 14.4 里面的 A、B 和 C 3 个部分就是最好的例子）
- 只有均值接近了，内部才容易匹配，不容易引起相互之间的牵扯和冲突，才能具有稳定的结构。

15

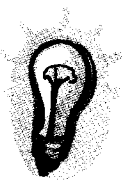
选择元器件，也有诀窍

15.1 尽量减少库存

检查库存，用库存的器件来设计。很多工程师拿到设计任务后，第一时间，就去上网找参考电路，找到后，开始采购器件。在设计的全过程，眼睛都盯着自己的项目。却比较少，先查公司的库存，看看是否有合用的器件。这样做至少有以下几个好处：

设计阶段的采购，量都是很小的。甚至很小批量采购时候，经常发生货款比快递费还便宜的事情。做过设计的工程师都知道，一般打给元件供应商的电话，第一就是告诉他们，你要什么。第二就是他问你，要多少。这个时候，即使你说“我要 500”。(其实你根本就要不了 500)，对方也经常会用一种失望的语调说“才 500 呀?”。碰上好一点的供应商，会卖给你，但是无一例外的，单价都提高了很多。碰上不好的，直接会告诉你：“我们的最小订量是多少 K，我们的采购周期是多少周”，这两种事情，其实都是让人很犯晕的事情。但是客观的说，这是经济规律使然，谁都改变不了。

仓库有同规格的元件，直接开领料单拿来用就好了。多快好省的途径！这个道理虽然简单，可是很多工程师就会忘记这一条。



减少库存的好处：

- 减少呆料。
- 加快采购周期。
- 减少不必要的采购。
- 采购周期快了。你的研发周期也快了。
- 采购买不到所需要的器件的时候，或者需要的器件很贵的时候，作为研发的你，一定耳根子不会清净。

- 将类似功能、而型号不同的元件统一成少数型号，有利于采购的时候获得一个好的价格。因为这样，每一个品种的需求量上升了，而品种总数量下降了。

且听我把这些好处慢慢道来……

15.1.1 代换法是减少不必要采购的有效途径

有时候，你要的和仓库有的，不会完全一样。比如，我以前用到一个电路（见图 15.1），其中有一个 NE555，以组成一个 10 kHz 的方波发生器，仓库有 NE555，但是却是 DIP 封装的，因为电路板空间的限制，只能用 SOIC 的型号。于是我马上想能不能用替代方案，我看见仓库里面有贴片的 CD4011，数量5 000 个，采购部说这是库存了 2 年的呆料。这个型号也是我以前经常用的，于是把这个部分的电路更改成图 15.2 的形式，输出频率同样调节成 10 kHz 的方波，效果是一样的。还消化了库存，避免了仓库采购新的材料。

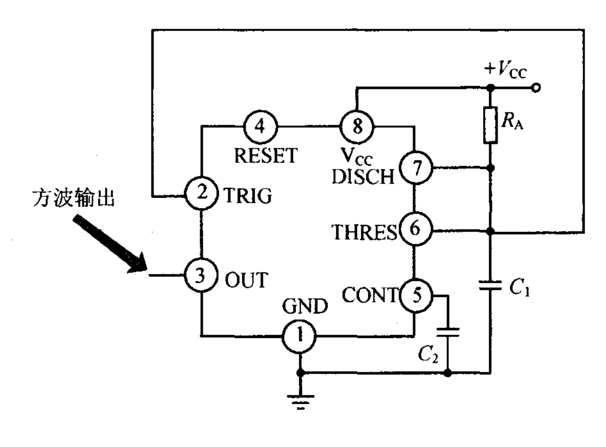


图 15.1

同样的，我们的设计中，很多三极管、二极管、运算放大器和电源芯片，各个公司之间都有类似的型号。只要明白其中的原理，看透器件的规格书，完全可以大刀阔斧的移植和变通设计。

比如三极管，如果三极管驱动的是很小的负载，比如图 15.3 中驱动 LED 灯的三极管电路，这个电路 LED 的发光电流很小，在 5 mA 左右。像这种场合用 9013、2SC1815 和 8050 等型号的三极管都可以，由于工作频率、电压和电流都很低，如果有多余的库存的话，甚至截止频率为 800 MHz 的高频放大的三极管

9018 在这里都可以安然工作。完全没有必要在一棵树上吊死。一般用三极管处理的信号，大多都是工作频率在 10 M 以下，电压在 12 V 以下的，或者干脆就是工作在开关状态的电路，这种情况下，三极管选择的余地非常大。因为现在的分立元件的三极管，几乎随便抓一个，集电极电流都能超过 30 mA。截止频率都能达到 100 MHz，对付小信号应用，绰绰有余。所以只要把规格书仔细看一遍，没把 PNP 和 NPN 搞错，一般很容易找到替代品。

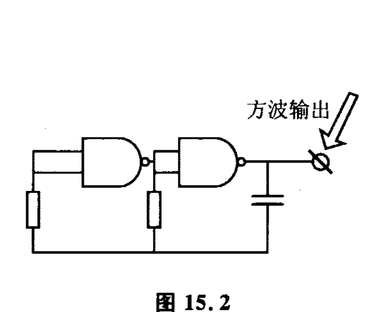


图 15.2

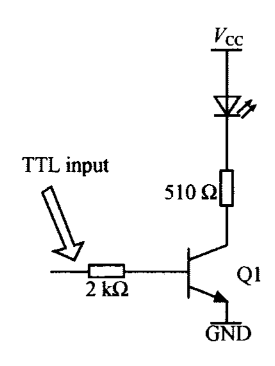
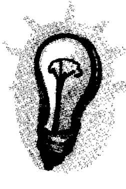


图 15.3

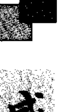
电源芯片也是这样，很多 3pin 的 LDO，其管脚排列都是一样的。仔细看看规格书，就会知道是不是可以替换掉你想要替换的型号。即使是 DC/DC 芯片，很多公司的产品也是 pin to pin 兼容的，替换也很容易。即使不能 pin to pin 兼容，只要还在绘制 PCB 的阶段，且 IC 能提供相同的输入输出，就可以简单替换。就像在图 15.2 中那样简单替换。根据我的经验，电源 IC 是所有 IC 里面，替换后产生麻烦最少的器件。

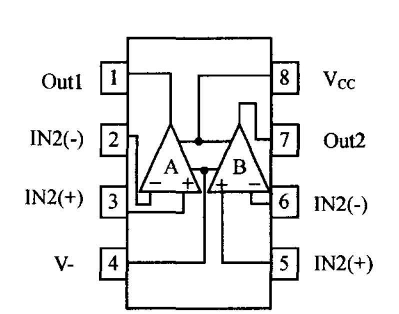
运算放大器也是一种很容易代换的器件，像双运放和四运放很多都是按照图 15.4 的 pin 规格封装的，这样就很容易找到 pin to pin 兼容的型号。运算放大器本身是一种适应性很强、很通用的器件。只要运算放大器的工作频率允许，基本都可以互换。替换后，电路基本不需要怎么改动。



以上的替换，都必须建立在以下基础上：

- 充分理解了所设计的电路。
- 充分阅读了所使用器件的规格书，对不同型号之间的差异是否会影响电路，有充分的了解。





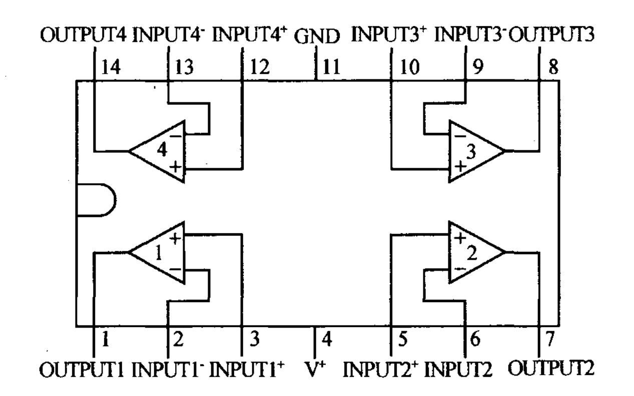


图 15.4

15.1.2 尽量统一电路板上的元件参数

执行这一条的理由和上面几个段落所陈述的中心思想是一样的，就是“减少库存!”。除此之外还会带来另外一个便利性：

如果电路板上的元件品种多，那么生产的时候，无论上贴片机、插件机还是人工插件都会增加工序。在工厂里面，工序的多少直接决定成本。有人设计的电路板，在贴片机上，安装了一个料盘，然后一个电路板只贴一个元件，如果这种情况是 IC 还可以原谅，如果电路板上有很多 RC 元件也这样的话，一般就说明设计者有问题了。因为在贴片机和插件机上，一般生产之前都会调试一下机器，会造成一个固定的元器件损耗，比如在机器准备要生产一个新的电路板之前，贴片机和插件机有可能会把料盘里面的头几个元件，用机械手或者压缩空气拿出来，然后在机器里面鼓捣一下后丢弃，以检查和校正机械手和机器里面的光学定位系统。为后面的大量生产做准备。这种情况在产量不高的电路板上，浪费尤其大。而且在这个个性化消费和商品过剩的时代，越来越多的产品的产量都是不高的。所以从这个意义上说，要尽量避免一个料盘在一个 PCB 上只贴一到两个元件的情况。

比如，有一个电路板，电路板上都没有 2 kΩ 和 510 Ω 的电阻，但是现在却要加入图 15.3 那样的驱动 LED 灯的三极管电路。这样电路板的 BOM 表中就多出了一个 2 kΩ 和 510 Ω 的电阻，但是电路板上现有的 BOM 中有 3 kΩ 4pcs、470 Ω 2pcs，这样，就可以把 2 kΩ 更改成 3 kΩ，510 Ω 更改成 470 Ω。而电路会照常工作，不会有用户可以察觉的变化。通过这样更改元器件参数，电路板变得

好生产了，元器件的数量也少了。

把这个问题扩展开来，其实相同的例子有很多。比如现在很多枪械使用的子弹，规格都是少数的几个型号，现在很多在服役的冲锋枪和手枪使用的是同一种子弹。同样的，很多机枪也使用步枪的子弹。很多型号的坦克和装甲车都会使用相同的模块，比如机车底盘、通信模块和火力系统等。比如一个坦克、一个装甲车同时在战场上受损，坦克损坏的是通信系统，装甲车损坏的是车辆底盘的机械系统，这个时候就可以把装甲车上同型号的通信模块拆下来，安装到坦克上，坦克立即就能恢复完全的战斗力。这样的设计可以节约军队后勤保障部门的运转成本，犯不着在军火库里面储备那么多款式的弹药和军械维修材料。在降低军队物流成本的同时，也降低了军火商的运营成本。

绝大多数航空公司在管理中，也遵循这个原理。一般一家航空公司都只会只用一个公司生产的飞机，不会一部分飞机是波音、一部分飞机是空中客车。统一成一个公司的飞机后，所有的飞机都只是少数几个型号，这样可以节约员工的培训成本，维修机师的经验也可以完全分享，管理流程更容易单一化、程序化。最重要的是还能降低维修配件的储备数量，因为飞机的维修配件都非常昂贵，少一点储备，就多增加了一点流动资金。

15.1.3 尽量选择供应商量产的产品和经典的型号

下馆子吃大餐的时候，如果不知道那个馆子口味好，通常有一种最简单的办法，就是谁家的馆子门口车多，就选择那个馆子。或者更简单的，挑选一个名气大的。其实选择元器件有时候和这个有点类似。

供应商和你一样，他的不同产品也有不同的产量。供货、价格和技术支持做得比较好的，一般都是他们出货量大的产品。这个也是所谓的“马太效应”，具体地说如果某个供应商，他的某个产品出货量足够大且产生盈利的话，他通常会把盈利重新投入到这个产品线上，让这个产品线做得更好，以产生更多的利润。所以这样的产品通常供货、价格和技术支持都做得比较好，终端用户和代理商也乐意参与。最后形成一个很强势的产品线。

对于硬件工程师来说，这样的产品线通常意味着某个 IC。这样的 IC，不仅 IC 本身有问题的概率很小，而且价格好且不容易断货。设计和生产的时候，碰到麻烦，去找原厂和代理商，都比较容易解决。而如果你不幸撞到一个销量不好

的 IC 的话,首先在设计的时候,肯定没有多少量,他也不一定会提供多少技术支持。生产后也不能得到一个好的采购价。

使用经典型号的 IC,也有很多好处。经典型号的 IC,比如运算放大器的 LM358、LM324;振荡器的 NE555、LM358;数字电路的 CD40 系列;TTL 74 系列;电源的 LM317、7805、78L05、MC34063、TL494、UC3842 等。这些型号的 IC 都是久经市场考验的,在市场上都生存了 5～8 年以上的型号。货源充足,价格也非常低。采购部门在采购这些 IC 的时候,容易处于主动地位。而且这些产品在市场上经过时间的考验,品质都很稳定。虽然这类 IC 名气太大,也有鱼目混珠的货色,但是通过定点采购,很容易解决这个问题。

15.1.4 有时候,浪费也是一种节约

有一次,仓库的仓管跟我聊起现在 IC 器件的库存品种太多,比如 8 位 CPU,有 AT89S52、AT89W55、MEGA8、PIC16C55、PIC16C57 和一些义隆的 CPU。而且即使一种 CPU,既有插件的又有贴片的。贴片的还分 SOIC 封装的和 TQFP 封装的。而且绝大部分器件的库存都不多,每次的需求量也不大。这样无疑将造成仓库管理难度的增加,不仅占地方,盘点发料都很麻烦,而且很多品种用量也不大,采购的时候也很被动。

后来我们也变得比较体谅采购和仓库,于是就改进了设计,对于那些产量不大、或者还在测试市场的产品,尽量用仓库现有的库存,即使仓库的库存元件比我的需求更高一些,也照样用。比如一个连接器,我只用到了 18pin,而别的产品大量使用 22pin 的连接器,那就用这个 22pin 的连接器,即使浪费掉 4 个 pin 也可以。而且像长度 20 cm 和长度 25 cm 的连接电缆,都统一成大量使用的 25 cm。

这里,看似设计的时候大材小用,浪费了资源,但是这样做,却可以节约小批量采购时的采购成本和物流成本。如果产量在一段时间后达到一定数量,再去修改 BOM 也不迟,这样做,虽然麻烦一些,却可以节约公司的成本。

当然,这个办法只是对于中小产量的产品,如果像 NOKIA 手机那样,动辄 100 万台的设计,这个办法还是不用为好。因为巨量生产的企业,采购都处于强势地位,不需要考虑这些。而且巨量生产,有多少库存都能消化掉。但是这样的强势企业实在比较少,相信看这本书的 95% 的读者设计的产品都没有巨量

生产。

15.2 配元件，要等于或者接近元件的额定值

做硬件，尤其是量产的硬件，经常会出现在实验室调试好的电路，一个批次生产下来就出现问题。这里面有很多是由于电路在设计的时候没有调试在最佳的参数，由于在实验室做样机，都只是少数几个板子，使用的元器件通常是一个批次的，本身元件与元件相互之间的误差就很小。等量产的时候，批次变化了，元器件的误差可能也会跟着变化。

举一个例子：LED 亮度变化。

我曾经设计过电话机，一般的电话机都有一个 LED 指示灯，指示听筒是否拿起来了。做样机的时候，在仓库里面找到了一个红色的直径 3 mm 的 LED，串联了一个电阻。连续做了 100 个样品的小批量，效果都很好。量产的时候，头 2 个月生产了 3 000 台都没问题。第 3 个月，车间报告："指示灯太暗了"。我到车间去一看，还真是这样。我们在车间忙碌了 4 个多小时，发现：

- 检查了电路板实物和原先的设计图纸，并没有发现什么差异。
- 把实验室里面剩余的 LED 样品换上，亮度立即恢复正常。
- 我们又跑到仓库检查仓库的出库和入库记录，发现上次 3 000 个没有问题的 LED 和目前使用的 LED 不是一个批次，也不是同一个厂家。
- 原先批次的厂商是广东东莞的 X 公司，现在的是广东惠州的 Y 公司。

同事说，这下好了，找到原因了，叫他更换批次和厂家就行了。可是我重新翻阅了一下仓库的账本，发现该 LED 的库存足足有 10 万个，问了一下采购部，得到的答复是："这个批次单价要比上个批次便宜 0.015 元，所以采购了这个批次。一下采购 10 万个是因为按照本厂今年的生产计划和任务，也只能用 3 个月而已！"

这么一来，还只能想办法把这 10 万个 LED 用掉。

回到实验室，把电路板重新测量，对照了新的 LED 和旧的 LED，发现它们之间除了有亮度的差异，没有别的差别（比如 LED 点亮时候的电压降和 LED 的工作电流等）。把广东惠州 Y 公司提供的 LED 规格书拿出来，发现它们的 LED 典型工作电流都在 5 mA 左右，而我们的产品，设计工作电流只有 1.5 mA。

我们尝试着调整 LED 的限流电阻，加大 LED 的工作电流，在加大到 3 mA

的时候，二者的亮度差异已经基本消失。但是电话机这样的产品，LED 的工作电流会影响到话路的特性阻抗。所以，在调亮了 LED 后又重新调整了电话机的电声参数，然后写了一个文档，指导车间更改电路板。

这样更改过后，无论用什么批次的 LED，都没再出现有亮度变化的问题了。

图 15.5 是某厂商的 3 mm LED 的简要规格说明，它们的测试结果都是在 20 mA 的状态下测得的。所以，后来我一般设计的时候，都将 LED 的工作电流设置在 8～10 mA。电话机由于要兼顾整机的特性阻抗，不可能按照 20 mA 来设计，但是我把它设置在 3.5～4 mA，尽量接近它的额定值，这样就不容易出问题了。

φ3.0*5.08mm Round with Flange LED Lamp（φ3.0*5.08毫米圆形有帽沿发光二极管）

Picture | Package Dimensions (mm)

Part Number	Chip Material	Lens Type	Emitting Color	Vf(V)If=20mA		λD(nm)If=20mA		IV(mcd)If=20mA		Angle
				Min	Max	Min	Max	Min	Max	2θ1/2
HF30BT04	InGaN/GaN	Water Clear	Blue	3.00	3.60	450	480	700	8000	15~20
HF30GT04	InGaN/GaN	Water Clear	Green	3.00	3.60	500	550	1500	16000	15~20
HF30WT24	InGaN/GaN	Water Clear	White	3.00	3.60	/	/	2000	20000	15~20
HF30RT04	AlGaInP/GaAsP	Water Clear	Red	1.80	2.20	620	660	600	9000	15~20
HF30AT04	AlGaInP/GaAsP	Water Clear	Amber	1.80	2.20	600	620	550	8500	15~20
HF30YT04	AlGaInP/GaAsP	Water Clear	Yellow	1.80	2.20	580	595	350	4500	15~20

图 15.5

15.2.1 配元件，不要超出元件的最大值和最小值，更不要超过极限值

图 15.5 中的发光二极管，如果在别的电路中，为了追求亮度，用图 15.6 那样的电路，4 个 LED 并联点亮，每个 LED20 mA，加起来就是 80 mA，这个时候，就不是什么三极管都可以用了，比如前面提到的 9018，就不能使用，因为它的集电极电流的极限是 50 mA。如果只驱动一个 LED 是很安全的，但是驱动 80 mA 的负载，就值得考虑了。图 15.7 是 9018 三极管的极限参数。

图 15.8 是 9018 三极管的一个额定参数，这个管子的最小直流增益可能会小到 28，正常产品的直流增益会在 28～198 之间。但是做设计的话，就要同时

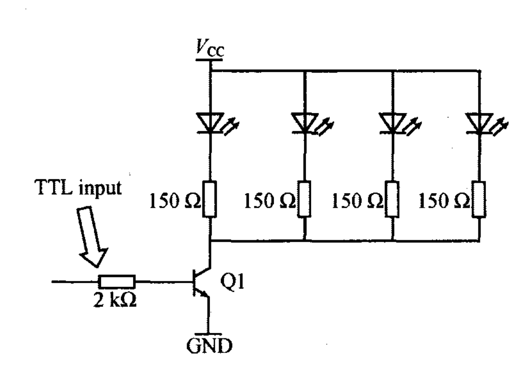


图 15.6

■■ 绝对最大额定值(T_a=25 ℃)

项　目	符　号	额定值	单　位
集电极—基极电压	V_{CBO}	30	V
集电极—发射极电压	V_{CEO}	15	V
集电极—基极电压	V_{CEO}	5	V
集电极电流	I_C	50	mA
集电极耗散功率	P_C	100	mW
结　温	T_J	150	℃
存储温度	T_{stg}	–55~150	℃

这里可能超标

图 15.7

■■ 电参数(T_a=25 ℃)

项　目	符　号	最小值	典型值	最大值	单位	测　试　条　件
直流电流增益	h_{FE}	28		198		V_{CE}=5 V,I_C=1 mA
集电极-基极截止电流	I_{CBO}			0.05	μA	V_{CB}=12 V,I_E=0

这个最小值要注意

图 15.8

考虑最大值和最小值。如果是 28 倍的直流增益的话,那给 9018 基极的电流就是 80 mA/28=2.85 mA。那图 15.6 中的 2 kΩ 电阻设计合理吗?计算一下,假

设 TTL 的 1 电平时 4.5 V，扣掉 9018 的 E－B 压降 0.7 V，那 $I_b=(4.5-0.7)/2\,000=1.9$ mA，这明显有问题。所以这个时候即使 9018 的集电极电流足够的话，这个电阻也应该更改成 1 kΩ。综合以上因素，9018 不适合。图 15.9 和图 15.10 是三极管 9013 的参数，这 2 个参数决定了，它可以适用图 15.6 的电路。

■■ 电参数(T_a=25 ℃)

项　　目	符　号	最小值	典型值	最大值	单位	测　试　条　件
直流电流增益	h_{FE}	64		300		V_{CE}= 1 V,I_C= 50 mA
集电极-基极截止电流	I_{CBO}			0.1	μA	V_{CB}= 25 V,I_E= 0

这个最小值就好一些了

图 15.9　9013 三极管的直流增益范围

■■ 绝对最大额定值(T_a=25 ℃)

项　　目	符　号	额定值	单　位
集电极—基极电压	V_{CBO}	40	V
集电极—发射极电压	V_{CEO}	20	V
集电极—基极电压	V_{CEO}	5	V
集电极电流	I_C	500	mA
集电极耗散功率	P_C	600	mW
结　温	T_J	150	℃
存储温度	T_{stg}	-55~150	℃

这个符合要求

图 15.10　9013 三极管的极限参数

16

果粉工作室 Retro Workshop

优秀设计的10大要点

1. 具有优秀的实用性，经得起用户挑剔的眼光

这句话是说，设计的产品要能达到预期目的、产品要能经得起时间、工作环境和成本限制的考验。在产品开始通电工作后，能对得起客户为产品所支付的费用。我见过一些新产品，市场宣传轰轰烈烈，产品却不怎么样。相反一些老产品，经过多年的设计修改和量产、批量使用，性能和价格都无可挑剔。

2. 有很好的稳定性，能够扛住很多恶劣的条件

作产品，完成后，还需要作一系列测试，比如温度测试、湿度测试、电压适应性测试和电磁兼容测试等。只有经过了这些测试，才可能让电路在出厂后，不至于因为碰到环境问题而死机、器件损坏等。我曾经见过以下几种要命的产品问题：

- 一个用MCU设计的机器在夏天运行良好，但是到了冬天，穿化纤衣服的用户操作机器的键盘，通过静电放电就能让机器死机。
- 在工业现场，操作设备的工人没有按照流程操作机器，导致逻辑死锁。必须复位才能恢复。这样的设计也是不可取的，必须有从错误中恢复的能力。
- 一个生产了多年的电话机，一向用户口碑良好。某天公司老总需要降低成本，命令把印刷电路板换成了便宜的纸基板，结果在南方潮湿阴雨天气下，纸基板吸潮而工作不正常。遭到大量用户投诉。

良好的稳定性是有代价的。首先设计者就要有足够的设计经验，在第一张图纸画下去之前，就要尽可能评估出产品可能遇到的使用风险，甚至生产过程中的风险。在设计的过程中，也要一步一步地消灭不稳定因素，最后才可能得到一个稳定的产品。

3. 好的电路通常都有很好的噪声抑制性，对噪声有必要的免疫力

同样是上述电话机的例子，在关键位置增加了元器件、增加了焊点之间间距

以及增加了抵御漏电的电路，这样改进后，即使在使用易于吸潮的纸基板时，一样能够稳定工作。

玩过收音机的都知道，超外差的接收电路要远远好过直放式电路和超再生电路。原因就是超外差的接收电路有一个变频的过程，而且将中频进一步放大和滤波。所以它的频率选择性在这三者之中是最好的。现在几乎稍好一些的无线电设备都是超外差的接收电路，比如收音机、电视机、手机和无线网卡等。有的设备为了进一步提高灵敏度和频率选择性，还会作二次变频处理。

电路在工作中，经常会有一些莫名其妙的干扰，比如电源上串入的干扰、电源纹波和毛刺。输入信号超出允许范围等。良好的电路对这些干扰予以尽量多的过滤，应该对这些干扰信号不敏感，才能保证电路的稳定持续运行。

4. 好的电路对输入条件有良好的适应性，即使有异常的输入，也能得到稳定的输出

一个电路的输入条件包括很多，比如电源的输入、视频和音频的输入、控制信号的输入、用户按键操作序列的不同输入等。比如电视机，电源只允许 220 V 输入的肯定没有 90～230 V 输入范围的好；全制式的肯定要比单一制式的好。

即使碰到恶劣的输入信号，电路也应该保持稳定。比如冬天穿化纤衣服的用户操作机器，不小心在电视机的视频输入打了一个静电火花，电视机不应该就此瘫痪。而应该在这种冲击信号过后保持原有的性能。

5. 好的电路灵敏度很好，信号哪怕再微弱也纤毫毕现

用电视机来说明这条理由，应该是不错的。电视机是一个影像和声音的还原设备，信号从天线端子接收下来，到最后在显示屏上显示出来，要经过多个通道。比如射频高放、变频、中放和视放等。要得到纤毫毕现的画质，必须要注意每个环节的设计。比如视放电路的高频响应不好，图像的细节就很差；视放电路的低频响应不好，图像的亮度和对比度就会出问题；中放部分如果带宽不够，也同样会影像图像的质量；如果高放部分电路的噪声系数没有得到有效控制的话，图像就有麻点……

手机中射频电路设计的不良，会导致后面的数字接收部分误码率大幅度上升，有效通信距离也会大幅度下降。为此手机可能需要加大发射功率和无线基站联络。表现最外在的现象就是：手机费电和容易掉线等毛病。

好的电路只在它自己的工作频率内，不会忽略掉微小的有用信号。而其他的频率，应该尽量排斥，这就是常说的抗干扰特性。因为自己工作频率内的信号

才真的是“有用的信号”。

以前有和设计雷达的工程师聊过飞机上的微波雷达。雷达高频部分的信噪比高的话，就能捕捉到很微弱的信号，可以提前几十公里到上百公里发现敌机。如果这个射频电路设计得不如敌机好的话，那就是敌机先发现我方的飞机。所以，一个电路设计得不好，也许可能要付出的代价就是飞行员的生命和一场战斗的失败。

6. 好的马儿跑得快，吃草少

这句话其实是说电子产品的低功耗很重要。

这个好理解，电子产品节能是硬道理，尤其是最近石油、煤炭等一次性能源价格节节高升的时代背景下。一个电子产品如果是市电供电的，就可以不考虑产品的功耗问题，至少不用放在第一位来考虑。因为，10 W的功耗和5 W的功耗，对于市电电网来说，没有太大的差别。而要遵照某些标准，降低功耗时，（比如电视机的待机功耗和手持设备的设计）即使不增加别的功能，在设计时要考虑的问题也会多很多。比如电路板上有很多单元，要节电的话，必须随时关闭这些外部设备。等重新要使用这些设备的时候，又要重新按照特定的时序上电，然后按照特定的顺序初始化它们，而不需要节能的话，只要在电路板上电之后，初始化一次就可以了，只要这个外设的电源没有被切断，就无需重新初始化。所以为了节约能量，设计的时候需要付出很多额外的精力来对能源进行管理。

其实这句话还可以引申为“好的电路消耗资源不多，可以节省资源”，这里的资源可以指：

- 消耗的电力。
- 电路的生产成本。
- 产品的设计周期（就是产品的时间成本）。
- 用户使用这个产品需要的代价（比如购买的价位以及产品人机接口是否易于理解等）。
- 产品是否便于维护（免维护最好）。

7. 好的电路可以衍生出更多好的设计

好的硬件和软件，一定是模块化很强的产品。模块化做得好，那每个模块的封装性就很好，对外的接口就一定会比较简单。有了这几条，在下一个设计的时候，就很容易将上次的设计重新拿来使用，而且拿来的模块可以免去很多重复的设计和测试工作。



在美国911之后，大家对电视画面中本拉登手持的AK-47冲锋枪都很熟悉了。其实苏联在二战时期曾经大量使用过一种叫做“PPSh(波波莎)”的冲锋枪，战争初期，德军的攻势势如破竹，苏联大部分的兵工厂被摧毁，而前线却迫切需要大量的武器装备，尤其是需求量最大的步枪和冲锋枪。在这种情况下，只有生产“最简单的结构、最经济的设计和最优良的火力”的冲锋枪才是上上之举。

1941年，PPSh冲锋枪诞生了，命名为PPSh41，在整个二战期间，PPSh41不停地被制造，直到战争结束时，约有500,0000支PPSh41装备苏联红军。PPSh41以其结构简单、动作可靠、性能优良、火力猛烈而且造价低廉而饮誉武器界。PPSh制造过程中大量使用冲压组件，焊接或铆接而成。拆卸保养非常简单，即使懒得保养也没有关系，枪管和枪膛内侧都进行了镀铬防锈处理，这在当时也是绝无仅有的。这些特点使其具有无以伦比的可靠性和耐用性，无论雨雪、污泥还是低温，都不能阻碍其倾泄弹雨。

而苏联后来的很多手持兵器，很多都部分兼容了PPSh41的设计，比如采用和PPSh41一样口径的子弹、同样的枪机等。这样不仅后续的产品设计周期的费用都大幅度降低，而且后续的产品在装备部队后，可以和先前大量使用的PPSh41互换部分配件，这样就会在无形中降低军队的后勤保障成本。

8. 好的电路朴实无华，并不一定要用昂贵的IC

这一点不需多说，请看看上面冲锋枪的例子，就能说明问题。

9. 好的电路需要由优秀的工程师来开发

一个设计，不论好坏，都是其设计者思想的现实表达。只有设计思想达到了某种层次，设计出的产品才能达到某个层次。而设计者每设计一个产品，只要他愿意总结，总可以提升自己的设计层次。

同样还是上面冲锋枪的例子，如果没有前线战士对武器耐用性的需求、没有设计师多年工厂批量生产的经验、对军队作战和后勤保障系统的了解和设计师执着的敬业精神，要想完成“波波莎”冲锋枪那样的经典设计，几乎没有可能。所以，无论是机械产品还是电子产品，都可以从产品中看出设计师的思维方式和经验水平。

10. 好的电路不容易设计

道理类似于上面的道理，做一个电子实验很容易，但是设计一个产品是相当不容易的事情。做实验只要考虑电路可否实现原先的构想，但是实现的质量如何，基本不用考虑。而作产品除了要实现产品的功能外，还要考虑以下的问题：

- 较低的成本。
- 易于生产。
- 较快的设计周期。
- 元器件易于采购，不会因为某个器件的供货中断而终止生产。
- 易于发现运行中出现的问题。
- 用户使用的便利性和易于理解性。
- 产品在恶劣环境下发生故障的能力。
- 设计文档在后续的设计中，尽可能的能够被再利用。

我们都知道20－80定律，优秀的设计也只有20%。在设计过程中，所作的一切努力都是让自己的设计属于那20%。而作为一个产品，就需要对照以上几点，如果有其中一点做不好，设计就可能会落入最普通的那80%中。



17

由数组非法操作想到的

17.1 一个地址越界的例子

请大家先看下面一小段程序：

```
void print_string(int str_ID)
{ char string[10];
switch(str_ID)
    {
    case 0: strcpy(string,"高压危险");
        Break;
    case 1: strcpy(string,"安全电压");
        Break;
    case 2: strcpy(string,"请切换到备用电池,然后给主电池充电!");
        Break;
    }
printf(string);
}
double calc_votage(int AD_result) //根据 AD 采样的结果计算修正后的电压值
{double ret,a,b;
    a = result;
    b = 10.244;
    if(result>100)  b = 12.577;
    if(result< 10)  b = 9.360;
    ret = a * b;
    return(ret);}
```

以上代码是我的一个单片机设计中的代码，大家能否看出以上的代码有什么问题？好像是没有什么问题，好，让我们跑跑。这时就发现了一个奇怪的现象，这个程序会随机的在 calc_votage 模块死机。死机的概率在 1/10 左右。正

是由于这种时有时无、低发生概率的故障，导致故障排查非常困难。

后来实在没招，只好用逐个模块屏蔽的办法，将整个程序的代码一部分一部分删去，最后发现，只要子程序 double calc_votage(int AD_result)存在，就会有死机的可能，但是用仿真器调试这个模块，却没有发现什么异常，CPU 进入这个子程序后，其中的变量和运行的流程都十分正常。而且死机发生的时候，CPU 并不在运行这个模块。令人百思不得其解。

百思不得其解还是得解，因为这是工作。由于这个故障来自于两个函数，而我又是这个函数的作者，估计某些东西我是很难找出来的。所以将这个代码提交给同事看。3 分钟后，我的同事告诉我，发现一个 bug，在函数 void print_string(int str_ID)的第二行：char string[10]；。后面一个语句 strcpy(string,"请切换到备用电池，然后给主电池充电！")；，其文字长度远远超过字符数组的长度。可能是故障的原因。而有时候不出故障，是因为字符串赋值没有超过这个变量的最大空间限制。

将语句 char string[10]；更改成为 char string[50]；后，程序再没有出过问题。

由于原先的错误，在执行语句 strcpy(string,"请切换到备用电池，然后给主电池充电！")后，strcpy 函数将字符串拷贝到 string 数组中，但是由于 string 的空间不够，所以字符串会向后面的地址顺延。而这些后续地址里面如果刚好储存了关键数据(比如其他的什么变量)，就会破坏这些数据，最后导致程序流程的紊乱而死机。

我原先之所以没有在断点跟踪中发现这个问题，在于 char string[10]这个变量的赋值语句本身不会直接导致程序紊乱，相反，在跟踪的时候，还可以看见这个变量被正确赋值。但是它完成复制后却破坏了 RAM 中紧挨着它的变量。

17.2 联想和感悟

这种问题，归根结底，是由于操作行为超过了自己的最大权限。对于某个数组的操作来说，其权限范围为这个数组本身。任何越界的行为都是非法的。仔细想想，类似这样越权的行为造成的故障还不少，比如图 17.1 所示的大家都见过的对话框。

出现这种问题，虽然不--定就是上面说的数组地址越界，但大多也是因为程



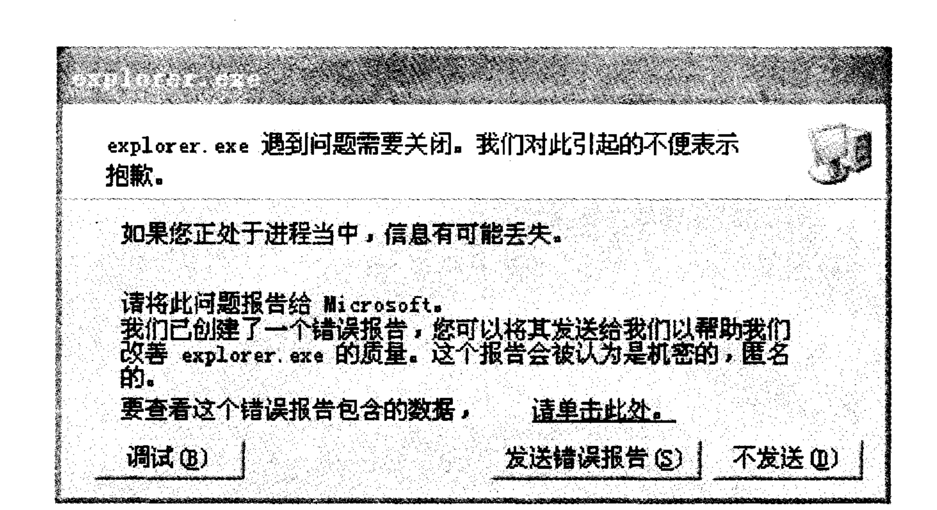


图 17.1

序执行了超越权限的操作，比如：

1. 在应用程序中，执行了 CPU 的特权指令。而绝大多数操作系统对特权指令都有管制。特权指令大多是操作系统和驱动程序才有权利执行。
2. 应用程序绕过驱动程序，试图直接访问底层硬件。而现在的操作系统，操作设备一般都要通过驱动程序来完成。不允许应用程序直接操作。即应用程序一般没有权限直接访问底层硬件。
3. 在多个程序并行运行的时候，A 应用程序的代码擅自读写 B 应用程序的数据。这也是被操作系统禁止的行为。在操作系统看来，只要是应用程序，其权限都是一样的。所以程序可以在自己的 RAM 区间中自由操作，但绝对不允许操作其他 RAM 里面的数据。即使只读不写也不行。
4. 访问了一个空指针。
5. 其他的可能。

以上几种可能，原因 1～4 都是程序执行了超过其最大权限的操作。尤其是对 RAM 的非法操作，大多可以造成灾难性的后果。所以在现在的很多 CPU 中（比如 ARM9 系列、ARM11 系列、X86 系列和 MIPS 系列）都有专门的模块管理这些情况，这就是 MMU（内存管理单元），它的功能除了调度内存以外，第二个主要功能就是像哨兵一样时刻监测以上的 RAM 非法操作行为是否发生，监视到了，就调度操作系统及时作出反应，防止问题进一步扩大。所以能跑多任务的大型操作系统，无一例外都需要用到 MMU。

以上这些管理措施，就好像我们现实生活中政府的管理架构。一个政府，我觉得可以理解为是一个国家的操作系统。一个政府，有很多下属机构，各个机构有各自的权限。有的负责金融、有的负责司法。如果金融部门对司法部门的工作不满意，必须告知自己的上级，让上级来管理司法部门。而不应该直接插手司法部门。这就是有序，就是不超越权限。

在做硬件设计的时候，也要注意硬件的各个模块互相之间不干扰。干扰无外乎以下 2 种情况：

- 一种是本模块的输出影响到了其他模块，比如射频干扰辐射到了其他模块、电源的不稳定(电压的晃动和纹波等)。这个也可以理解为是一种越界，要么就是这个模块的输出信号到达了不该到达的地方。要么就是模块的输出在包含需要的东西的同时还有不需要的东西(比如电源的纹波是不需要的，而 DC 电压是需要的)。
- 另外一种是本模块受到了来自于其他模块的干扰，或者输入信号中除了需要的信号以外还有不需要的信号。比如音频线串入的工频干扰。这种干扰至少有一半是可以通过抗干扰设计去除掉的。所以设计硬件的时候，要多想想，除了需要的信号，有时还会碰到不需要的信号？如果有的话，用什么对策去除或者弱化它的影响。同理在设计软件的时候，也要多想想，除了需要的输入数据，时候还会碰到不需要的异常输入数据。

18

废手机改成充电器



这个文章是我前几年写的，觉得这类 DIY 文章，对启迪大家的思维，还是有点用的，所以虽然过了几年了，也还是拿出来献丑了。

春节前买了一个数码相机，不想却是一个电老虎。刚买的南孚“聚能环”电池，才拍了 15 张左右就不行了。而原机为了降低售价，不为机器搭配充电电池和充电器而只作为选购件销售，且价格不菲。没办法，只好到电子市场买了 4 个大容量(1.2 V/1 500 mAh)的品新电池，如图 18.1 所示。

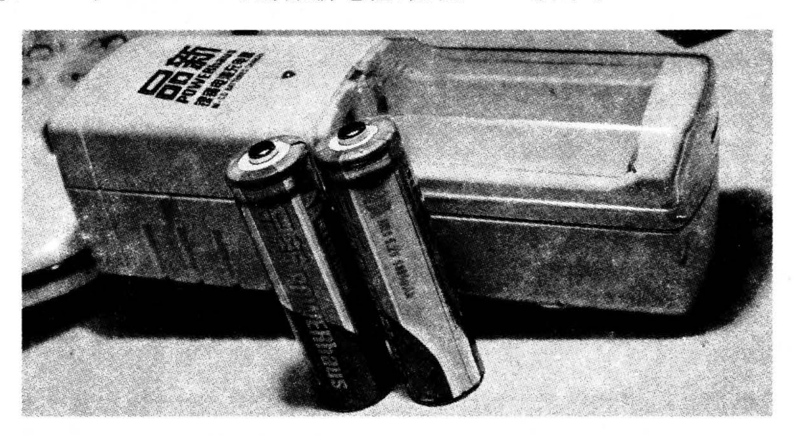


图 18.1

图中电池后面的充电器是我原先买的一个镍铬电池充电器，充电电流只有 80 mA，只适合于充 600 mAh 的镍铬电池。用来充这样的高容量电池需要 29 个小时左右。这样的时间如果在家还可以，要是出门在外，这么长的充电时间绝对令人难以接受。且它是一个低成本的简单充电器，没有过充电保护，长时间充

电容易伤害电池。

在电子市场上象 GP 那样品牌的智能充电器价格至少要 180 元以上。便宜的充电器也很多，只是都没有电池充饱指示，也没有过充电保护。假如你精通电子学，自己做一个当然是最好的。但即使是这样，如果加上人工成本，绝对不上算。

我突然想到了使用闲置的手机来改装成充电器，因为手机内部的充电电路一般都有专用的充电控制 IC，有完善的电池保护功能和充电过程控制功能。为此我在抽屉里找到了这个老古董，如图 18.2 所示。怎么样，这款 ERICSSON 手机相信大家不会陌生吧，虽然其块头比不上当年的砖头机，但相对于现在的手机绝对算得上是个大个子了，也就是因为这样，谁也不喜欢它，即使它没有损坏。



图 18.2

选择手机还有一些讲究，这个手机的电池标签和手机的充电电源适配器如图 18.3 所示。

它的电压是 4.8 V，容量是 650 mAh，这个电压刚好够我连接 4 个镍铬电池或者镍氢电池。这里强调一下：你可以选择其他型号的手机，但是手机的电池电压如果是 N×1.2 V 的，那只能够一下给 N 个电池充电，不可以擅自增加或者减少电池个数；也不能给额定单体电压不是 1.2 V 的电池（比如碱性电池和普通一次性电池）充电。否则手机内部的充电控制电路会按照错误的电压来进行充电控制，当然充电的结果也是错误的。2003 年以后的手机，配备的电池几乎清一色都是 3.7 V 的锂电池，这种手机或者适合来充 3 个或多个镍氢、镍镉电池，或者是拿来充单个的片状手机、数码相机的锂电池也是相当不错的。



图 18.3

然后去买一个电池夹，如图 18.4 所示，只要一块钱人民币。

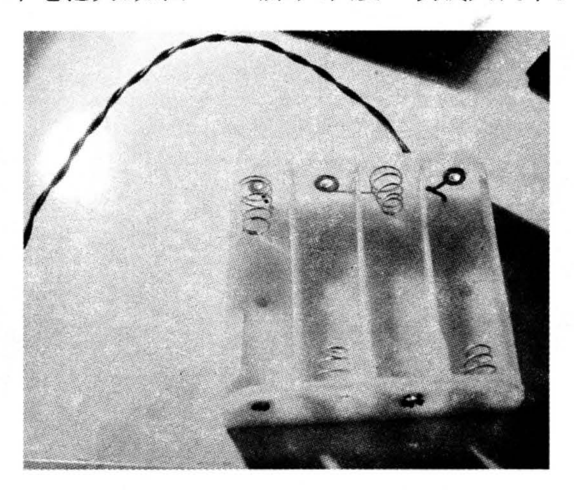


图 18.4

按照原机所配备的电池按照极性把电池盒的 2 根线焊接到手机后背的接线端子上，检查无误后在电池盒后面粘上厚的双面胶，然后将 2 者粘起来，如图 18.5 所示。

最后把充电电池安装到电池盒上，连接好手机的充电电源适配器，就可以开始充电了，充电过程中，手机的 LCD 显示屏会有动画的充电指示，等显示的电池图标停止闪动，电池就充好了。

这里需要说明一下：在图 18.6 中使用了 2 种不同的充电电池，这里只是在



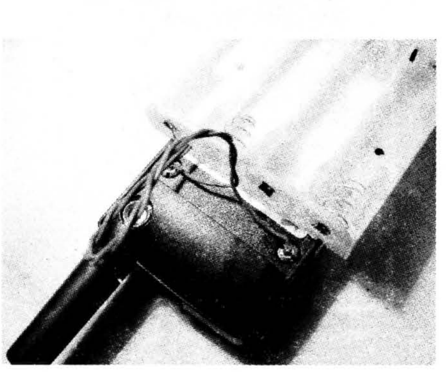


图 18.5



图 18.6

做这篇文章的示意，请大家在真正充电的时候一定要同时使用相同品牌和相同批号的电池，不要混用！

好了，充电器做好了，现在给 4 节 1.2 V/1 500 mAh 电池充电只要 5 小时。改造后的手机还是可以用的，你接上电池，在图 18.7 的 SIM 卡座中插入一个 SIM 卡，电话照样可以打。虽然这样的手机显得有一些异类，但好歹还是一个手机，你说是不是？





图 18.7

19

使用 Keil C51 RTOS 开发蓄电池监控器

19.1 RTOS 的优点

我们的客户向我们提出需要一种可以监控通讯机房蓄电池组的电池监控器，需要采集无人值守机房蓄电池单体电压、总电压、电流、温度等数据，可以输出告警信号、可以通过 modem 向远端提供数据和在机房中通过 RS232 接口配置数据。图 19.1 是该电路的原理框图。

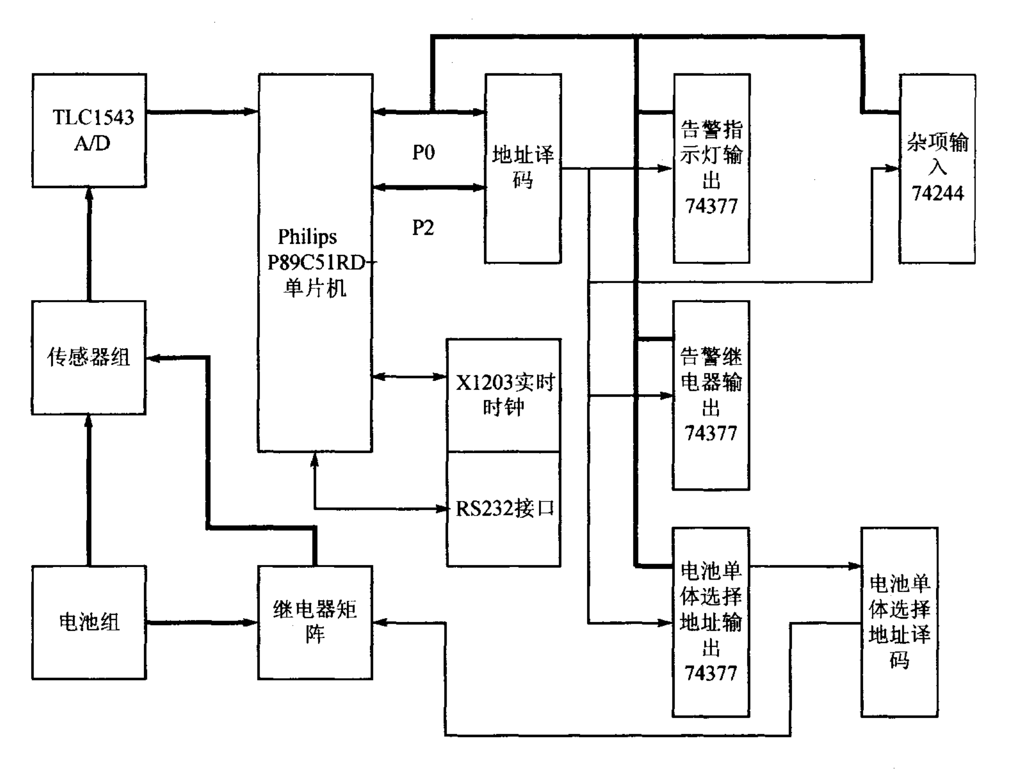


图 19.1

由于该系统中各个子任务需要及时响应，所以一般的程序块顺序执行的方式不适合于本系统。如图 19.2 所示，程序块顺序执行方式的最大难点在于需要控制每个程序段的执行时间，否则对一些事件的响应将丧失实时性。一些任务，如和 PC 机超级终端通讯，如果每次都只执行一个小时间段后都要退出，将导致软件结构复杂化，而且一个程序段在输入不同数据的情况下，执行时间往往是不同的，这样的情况下，也要保证程序都能够定时退出则更加困难。

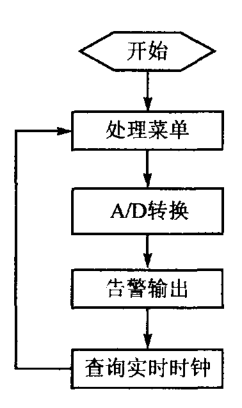


图 19.2

基于以上原因，我们考虑使用 keil C51 的单片机实时操作系统 RTOS，它不象 unix、windows 系统那样包含内存管理、任务管理、网络管理、文件管理等模块，它是一个非常简化的操作系统，只有任务管理和简单的内存管理，其它的模块都没有。该操作系统最多可以支持 16 个任务同时运转，而对于绝大多数单片机应用来说 16 个任务已经绰绰有余。以下是该操作系统提供的主要系统函数：

isr_send_signal	从一个中断服务程序中发送一个信号给一个任务
os_clear_signal	删除一个已经发送的信号
os_create_task	建立任务，把任务加入执行序列
os_delete_task	删除任务，把任务从执行序列中删除
os_running_task_id	返回正在运行的任务的任务号
os_send_signal	从一个任务发送信号给另一个任务
os_wait	等待特定的事件

19.2 本系统的软件结构

本系统使用 5 个任务来构建，如图 19.3 所示。其中主任务完成 4 个子任务的管理和协调；PC232 菜单任务完成和 PC 机上超级终端通讯的人机接口(同时还可以完成和 modem 的通讯)，通过该人机接口，可以让用户完成数据设置，也可以在生产的过程中配置、调试机器，增加生产和维护的方便性；数据采集任务完成电池总体电压、电流和电池单体电压的数据采集；告警任务根据采集来的数据计算出数据是否超标，并且输出相应的告警信号；实时时间任务完成管理 I^2C 总线的实时时钟芯片 X1203 的访问，并且实时刷新单片机 RAM 中的时间缓冲区，这个时间缓冲区的数据可以让别的任务共享。

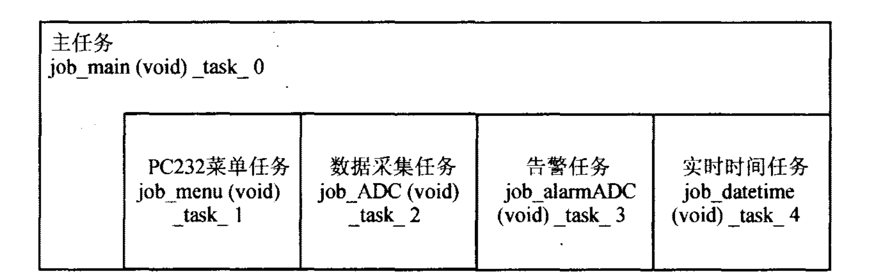


图 19.3

keil C51 的单片机实时操作系统 RTOS 其实是一个准并行的多任务环境，其根本原理和 windows 的多任务类似，就是将 CPU 的机时等分切割成为一些时间片，然后分配给每个任务，分配给某个任务的时间耗尽后，如果该任务还没有计算完毕，则强行收回 CPU 控制权，保护现场后将 CPU 控制权交给下一个任务，如此循环往复，周而复始。在微观上每次只有一个任务占用 CPU，但是由于时间片时间很短(11 MHz 下，大约每个时间片为 20 ms)，所以从宏观上看，各个任务是并行运转的。

当一个程序段按照多任务的方式来设计后，可以将每个任务都设计成无限循环，可以反反复复地查询自己所管理的的那部分硬件。也可以反反复复地计算数据，不用担心自己独占 CPU 后，别的任务会因为不能得到 CPU 机时而丧失实时性。而且当一个任务数据处理完毕后，不需要再占用 CPU 的时候，可以执行一个 OS_WAIT( )来释放 CPU，以便别的任务可以更及时地得到处理。以下是 2 个任务的范例代码，其中任务 0 是上述的主任务，它负责建立和删除每个子任务。任务 1 是上述的和 PC 机上超级终端通讯的人机接口的任务，任务 0 利用了 RTOS 的建立和删除任务的功能，时刻监视菜单任务，可以实现菜单很久没有人操作后自动从菜单树中的任意节点返回到菜单根节点。而这种操作不使用多任务来实现的话需要在菜单任务中插入许多代码，导致结构复杂化。

```
void job_main (void) _task_ 0
{unsigned int menu_time_out; long MLN_moni; //监视 MLN 是否发生变化的变量
   beep_lamp_test(); //测试指示灯和蜂鸣器
   system_init();  // 系统上电初始化
   os_create_task(1); //建立菜单任务
   os_create_task(2); //建立 ADC 任务
   menu_time_out = 0;
   MLN_moni = MLN;
```

```
    while (1)
     {os_wait2(K_TMO, 20);   //20个时间片周期才运行一次
      if(MLN = = MLN_moni) menu_time_out + + ;
      else {menu_time_out = 0; MLN_moni = MLN;}
      if (menu_time_out>menusystem_timeout_time) //如果菜单任务超时
         {os_delete_task(1); //删除菜单任务
          os_create_task(1); //再次建立
          menu_time_out = 0;
         }
     }
}

void job_menu (void) _task_ 1
{   init_comm_model();         //初始化串行通讯口
    show_copyright();          //显示版权、版本
    while (1)
     {proc_menu_talk();        //处理菜单
     }
}
```


19.3 多任务环境下的资源冲突

在使用 RTOS 的时候，要注意和一般的单片机软件系统不同。以前的单片机系统是单任务系统，每个子任务是顺序执行的，一个正在运行的程序段可以拥有和控制单片机内外所有的资源，可以任意安排使用。而在多任务环境下就不同了，资源在访问时如果处置不当，可能会发生冲突或其它不确定性，这表现为一些以前在单任务下可以良好运行的代码段到了 RTOS 环境下就不能得到正确结果。以下以本系统中 2 个访问 I²C 总线的任务为例子来说明这个问题。

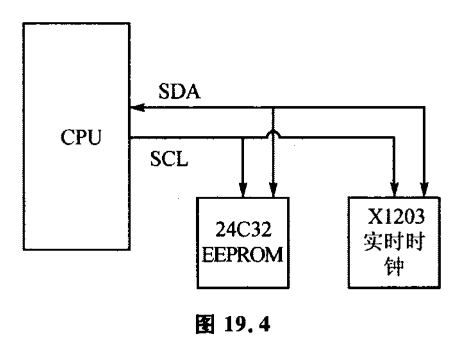


图 19.4

有 2 个任务，一个是前述人机接口中访问 I²C 总线上的 EEPROM 器件 24C32 的，另一个是实时时钟任务访问 I²C 总线上的实时时钟 X1203 的，这两个器件都挂接在 SDA 和 SCL 两个信号线上。请看图 19.4 所示的本系统 I²C

总线硬件结构图。

访问它们的时候，需要按照器件的时序要求来送出脉冲信号，才能完成 CPU 和器件之间的通讯。

图 19.5 是 I^2C 总线的 SDA 数据线上的波形图，如果按照本图上半部分的时序，I^2C 总线可以正常访问，但是假设在本例中任务 1 运行到黑竖线的地方，RTOS 发生了任务切换，任务 2 同样需要访问 I^2C 总线，于是它也向 I^2C 总线发送一串脉冲，虽然每个任务访问 I^2C 总线的时序都没有错误，但这时 SDA 数据线上的逻辑状态被破坏却是实实在在的。在这种情况下任务 1 和任务 2 谁也不能得到正确的数据。这就是多任务下资源冲突的一种表现形式。其实解决这个问题也不难，我们可以设置一个信号量来表示 I^2C 总线是否被占用，如下是它的代码：

```
bit IIC_bus_busy                    //I²C 总线忙标志,1 有效

while(IIC_bus_busy)                 //如果 I²C 总线被占用,则等待 I²C 总线不被占用
  { delay( ); }
IIC_bus_busy = 1;                   //锁定 I²C 总线
accessing_IIC_bus( );               //访问 I²C 总线
IIC_bus_busy = 0;                   //解锁 I²C 总线
```

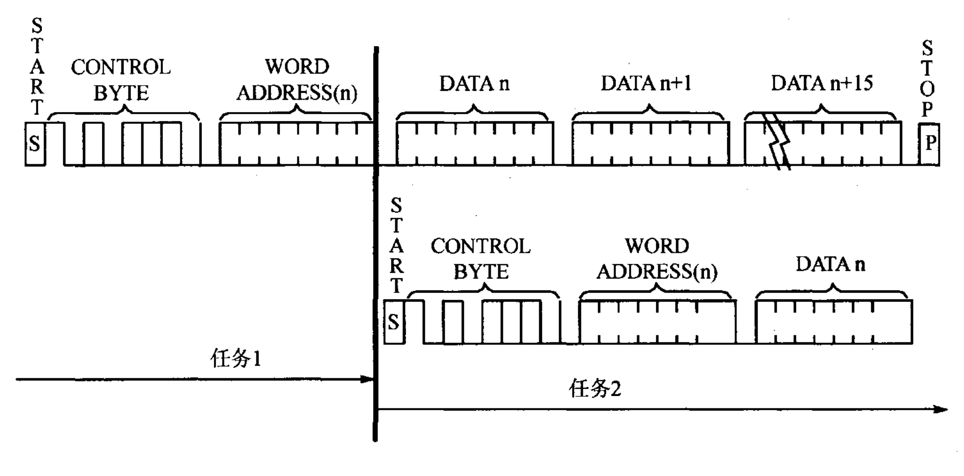


图 19.5

19.4 多任务环境下的函数重入

RTOS 的另外一个问题就是重入，所谓重入就是说一个任务正在运行一个函数，当 RTOS 发生了任务切换后，如果也调用这个函数，就会破坏上一个任务调用此函数的现场，导致上一个任务的计算错误。看以下的例子，这是本系统中用到的一个计算一个整数是 10 的多少次方的函数。(例如 $9=9\times10^0$ 返回 0、$789=7.89\times10^2$ 返回 2)。该函数定义了 3 个内部变量，假设任务 1 运行到 for (i =0;i<8;i++)这一行发生了任务切换，第 2 个任务重新初始化这个函数的内部变量，由于 C51 编译器通常情况下是将变量定址分配在 data 区和 xdata 区，这样任务 1 的运行状态就被破坏，导致计算错误。解决的办法就是加入一个关键字 reentrant 到函数声明行 char get_10n(long num)，把它更改成为 char get_10n(long num) reentrant，这样它的内部变量就在堆栈中分配，不会导致上述现象发生。不过由于 895X 系列单片机的堆栈深度有限，此方法会增大堆栈消耗，除非必要，能不用就不用。

```
char get_10n(long num)
{char i; long n,s;
   n = 1;
   if (num> = 0 && num< = 9) return(0);
   for (i = 0;i<8;i + + )
   {
     s = num/n;
     if(s = = 0) return(i - 1);
     else n = n * 10;
   }
   return( - 1);
}
```

解决了多任务下的资源冲突和重入后，RTOS 就比较完美了，相信大家在熟练使用 RTOS 后一定能够感受到它带来的稳定和高效。

20

果粉工作室 Retro Workshop

PCB 软件不为人知的技巧　Net Class

PCB 布线软件的书籍和资料大家应该都看得不少，网上有很多布线技巧的文章，大都是教人如何避免干扰，如何走地线的等，其实这些软件里面还有一个功能，也很好用的，只是绝大部分的书籍都没有介绍。这就是 Net Class 功能。

PCB 文件首次加载网络表的时候，没有对其进行分类。这个功能可以人工将无数的网络连接分门别类，比如分成 power、data_bus、Address_bus 和 Hi_volta 等类别。这样分类后可以分别对不同的类别施加不同的布线策略。

好了，现在让我们尝试一下这个功能（以 protel 为例）：

- 首先打开一个 PCB 图。
- 选择菜单“Design→Classes..”跳出图 20.1 的画面。
- 这里我已经预先定好了几个 Class，其中“All Nets”是 protel 默认的类别，这个类别包含了所有的网络。如果定义了布线规则，默认就是针对这个类别的。

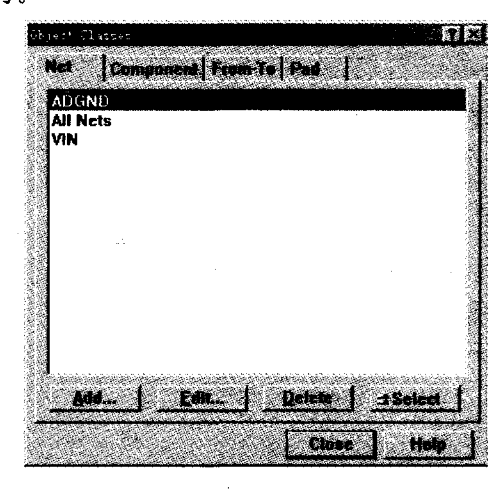


图 20.1

现在我要为这个 PCB 增加一个表示 CPU 地址总线的类别“Address_bus”，按下图 20.1 画面中的“Add”按键，在图 20.2 的画面中输选择“A0～A19”，然后选择“＞”把这些网络放置到右边的子窗口中，如图 20.3 所示。这样就建立好了一个新的 PCB 网络类别。用同样的办法，再建立“power”、“data_bus”等网络类别。

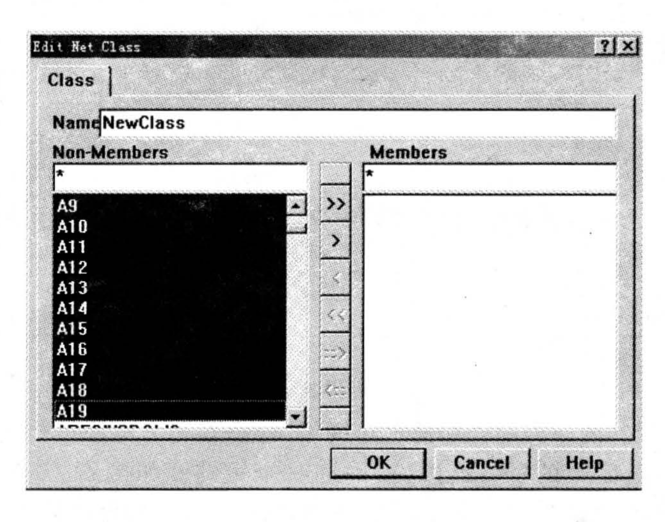


图 20.2

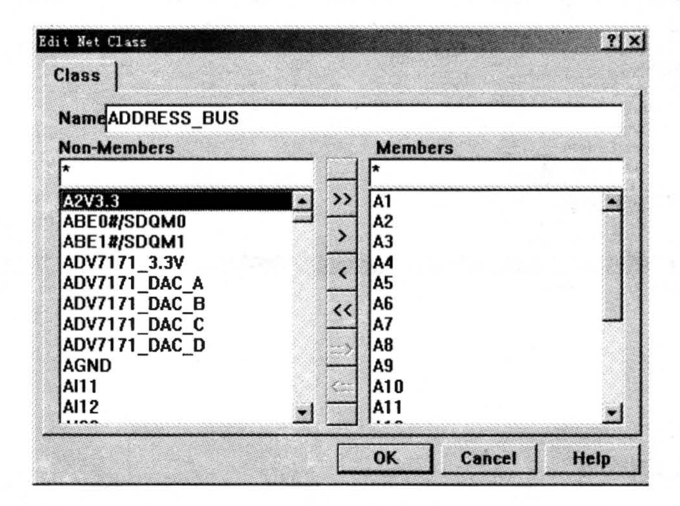


图 20.3

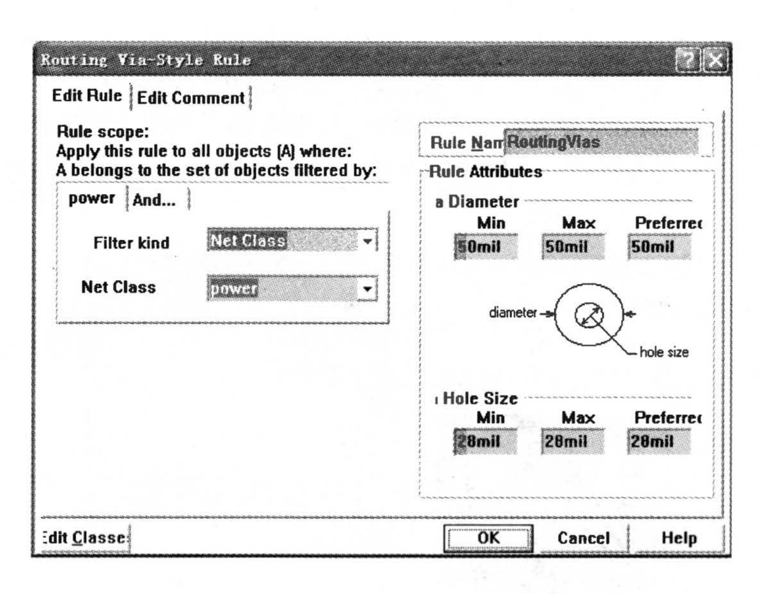


图 20.4

下面，为这些网络类别分别指定布线策略吧，首先为电源类指定布线策略。按下图 20.5 画面中的“Add”按键，增加一个策略。如图 20.6 所示，“Filter Kind”选择“Net Class”，“Net Class”选择“POWER”，然后可以分别设定它的线

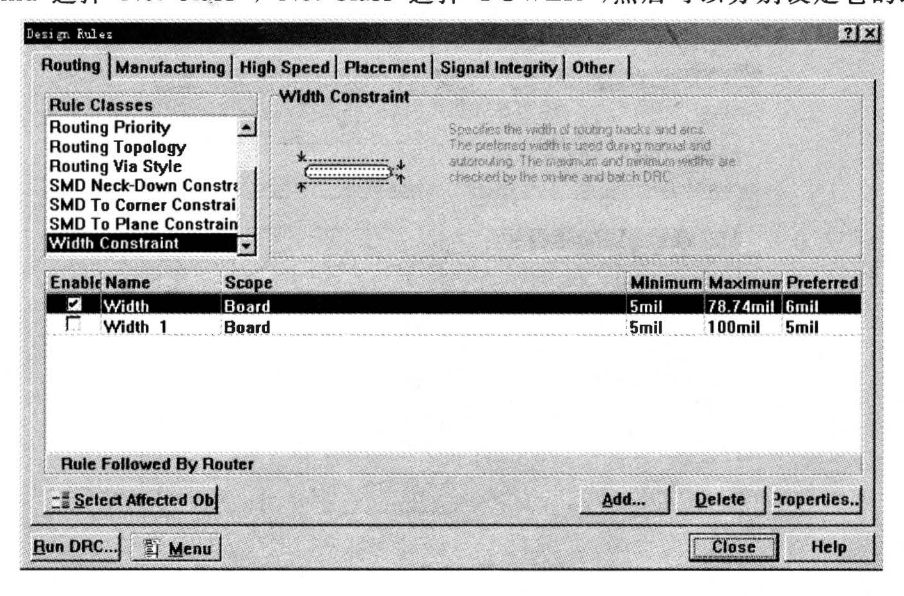


图 20.5

宽等参数，还可以为 POWER 类增加一个靠近限制的规则。（由于这个电路板是 4 层板，我这个工程就不设置靠近规则了。）

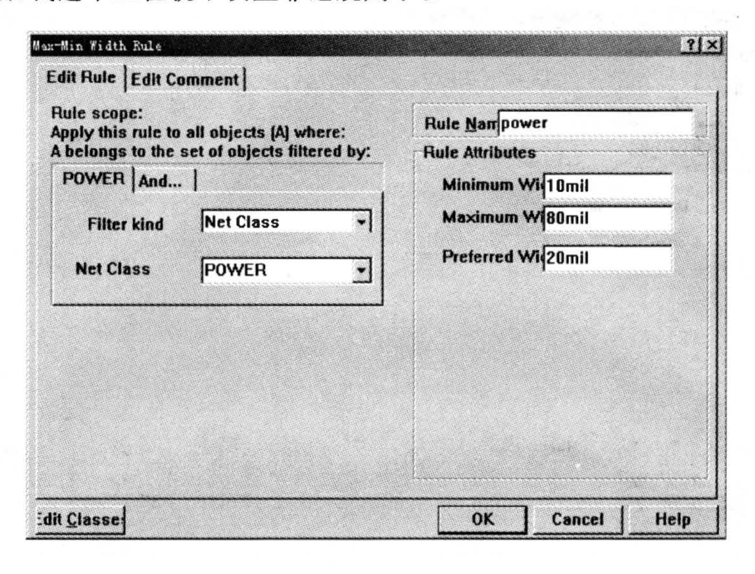


图 20.6

还可以指定电源网络的过孔大小。让电源线的过孔比一般的网络的过孔要大一些。因为电源有可能有 3 A 的峰值电流和 1 A 的平均电流。所以除了铜箔要宽一点以外，过孔也要大一些。可以为电源地 NetClass“Power”专门指定一个布线规则。这样设置好后，无论是自动布线还是手动布线，布线器放下去的过孔都要比默认的过孔大一号。

由于我的这个电路板是一个高速的嵌入式系统，CPU 外部总线频率大约为 200 MHz。所以地址总线和数据总线的设计就变得至关重要。每个地址总线相互之间的长度差不能太长，否则会造成传输延迟。但是手工去测量长度差实在麻烦，这里就可以为地址总线设置一个布线规则。

在图 20.7 的布线规则窗口中，选择“Length Constraint”，弹出图 20.8 窗口，选择“ADDRESS_BUS”类，可以设置总线最大长度和最小长度。图 20.9 所示窗口可以为这个 Net Class 设置蛇形布线规则。

设置了以上规则后，无论是手动布线还是自动布线，都会简单很多。在手工布线和修整电路板的过程中，不用再考虑这些参数了，因为你犯规后，PCB 编辑器会给出警告。通过这样的设置，可以一次为一大把信号线设置规则，不再需要一个一个信号单独设置了。可以节约不少时间，也可以避免很多不该犯的低级

错误。

图 20.7

图 20.8

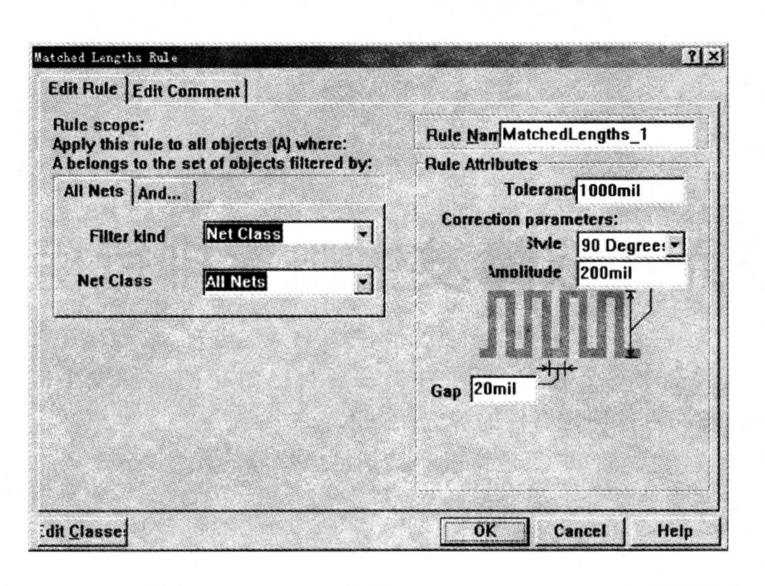


图 20.9



提示：

在布板的时候，很多低级错误都是可以避免的。但是都靠人工去记忆、去校对、去防止，那布线的人会累晕倒。把您的一部分想法，告诉机器。让机器替你执行和检查。这也是降低设计 bug 的一个诀窍。

21

电子元件故障发生概率排行榜

电子元件有十几个种类，各个种类都有自己特定的故障发生概率。如表21.1所列，来说明各种元器件的脾气。

本章所说的“失效率”都是特指产品正常出厂后的元件失效。不包含设计过程因为设计不完善所导致的元器件失效。

表 21.1

失效频率	器件的种类	容易失效的原因与后果	降低失效率的对策
10	镍镉电池	深度过放电后会导致电池不可恢复的损坏，所以长期储存不充电很容易损坏。过充电也经常会因为高温损坏电池。除此之外，它还有记忆效应，会导致容量的降低而失效	要定期充电，且要使用品质优良的充电器
10	镍氢电池	深度过放电后会导致电池不可恢复的损坏，所以长期储存不充电很容易损坏。过充电也经常会因为高温损坏电池	要定期充电，且要使用品质优良的充电器
10	普通一次性纸包装电池	包含一般的锌锰电池和碱性电池，在过放电后还长期放置的话，电解液会透过包装溢出，损坏外围的电子设备	定期检查电池的状况，可能的话，电路中增加欠压提示
9	锂电池	深度过放电后会导致电池不可恢复的损坏，所以长期储存不充电很容易损坏。过充电也经常会因为高温损坏电池	要定期充电，且要使用品质优良的充电器
9	灯泡	常见的就是灯丝断了；不常见的，就是灯泡漏气了	二战时期的电子产品很多使用灯泡作指示灯和限流电阻。二战结束六十几年了，所以新的设计尽量避免使用灯泡

续表 21.1

失效频率	器件的种类	容易失效的原因与后果	降低失效率的对策
8	高压大电流半导体器件	由于高电压大电流的存在，大部分失效的器件都是被高压击穿 PN 结	器件降额使用
8	普通一次性金属皮包装电池	包含一般的锌锰电池和碱性电池，在过放电后还长期放置的话，电解液会透过包装溢出，损坏外围的电子设备。不过其耐久的时间比纸包装的一次性电池好得多	定期检查电池的状况，可能的话，电路中增加欠压提示
6	有刷电动机	由于换向器的存在而导致火花，最常见的是换向器烧蚀，导致噪声和耗电同时上升。换向器烧蚀后又经常干扰其附近的电子设备。换向器烧蚀严重后，电动机就彻底不转了，形同废铁。但它的优点是比无刷电动机便宜，所以广泛使用在玩具等低挡设备中	选用品质好的电动机，不要超电压、超扭矩使用。超电压、超扭矩最容易导致换向器提早老化
6	硬电线(直径较细的单芯电线)	一般的故障是折断，在焊点的根部折断	尽量减少折叠的次数和幅度
5	大电流继电器	一般是超过额定电流使用，驱动感性负载的继电器故障率高一些	不要过流，感性负载要适度降额使用
5	大电流连接器	一般是因为接触电阻太大而烧蚀，或者超过额定电流使用	不要过流，所有连接器都要拧紧螺栓，以降低接触电阻
5	高压包	既然有高压，它的失效率一定不低	选用品质好的器件。到有品牌的厂商那里定点采购
5	可调电容	劣质的可调电容经常出现滑动触片接触不良的情况	选用品质好的器件，不过再好的可调电容其失效率也是大大高过固定电容的，所以，不用为上策
5	可调电阻	劣质的可调电阻经常出现滑动触片接触不良的情况	选用品质好的器件，不过再好的电位器其失效率也是大大高过固定电阻的，所以，不用为上策

续表 21.1

失效频率	器件的种类	容易失效的原因与后果	降低失效率的对策
5	小电流信号连接器	如果继电器有轻微的接触不良，流经继电器触点的是电流和电压都很小的信号，那继电器触点的微弱电阻变化都容易被弄到信号中去。比如这个时候传递的是音频，那故障发生的时候，经常能听见输出端有“沙沙”的杂音；如果传送的是视频，则有可能会有随机的横纹	电流和电压都很小的信号，尽量不要用继电器。而用电子开关比较适合
4	带 flash 的半导体芯片	大部分失效都是因为内部的 flash 信息丢失，其实这个时候，器件大多并没有物理性的损伤，大多重新编程一次就好了。扣除这个因素，其失效率和一般晶体管和半导体器件相当	注意电路板的 ESD 设计，也要注意使用环境的 ESD。这样就可以很大程度避免 flash 信息的丢失
4	电解电容	一般有 2 种失效的情况：一种是过电压或者反接导致炸裂，一种是电解液干涸（一般是高温环境才能出现）。电压越高、容量越大的电解电容出现故障的概率越高	反接导致的炸裂一般出现在设计过程中，量产的产品极少发现。还要注意电路板的散热和适度降额使用器件
4	高密度连接器	连接脚多了，出现接触不良的概率当然也大。比如 PCI 连接器、内存条连接器等。使用环境如果不够干燥、灰尘多也会大幅度降低连接器的可靠性	使用品质优良的器件，保持使用环境的清洁。出现故障的时候，用专用清洗剂清洗触点
4	无源晶体	不起振是最常见的，还有频率误差偏大也很常见。还有一种故障是频率跑到标称频率的 1/3，比如标称 27 MHz，结果跑到了 9 MHz	选用品质好的器件
4	纸基线路板	强度最低，一般是单面板。板材较脆。抗潮湿环境性能最差。很多售价很低的电路板在潮湿天会出故障，都是因为使用了这种电路板。这种电路板铜皮和板基的附着力较差	如果非要使用，注意线路间距要足够大。焊点的面积要足够大





续表 21.1

失效频率	器件的种类	容易失效的原因与后果	降低失效率的对策
3	CRT 显像管	虽然又大又笨重，但是它和磁头一样是很皮实的器件。最常见的老化是散焦。其次是其内部的三基色荧光粉的老化速度不一样，导致偏色	如果偏色，重新调整颜色就行。如果散焦，暂时提高灯丝电压虽然可以对付一阵，不过那说明显像管真的阳寿快尽了
3	LCD 玻璃片	这一般是那种廉价电子设备的 LCD 显示屏。大部分的不良是玻璃片和 PCB 连接的部分不良，比如斑马条和斑马纸。还有就是 LCD 受冲击碎裂	
3	LCD 显示屏	基本是坏点或者背光不良。比较恶劣的坏点是整行整列的坏，或者损坏一整个矩形区域	如果只是坏点，将就着还能用，如果是整行整列或矩形区域的损坏，就没招了
3	电源变压器	一般短时间的过电压和过电流，都不至于损坏变压器，但是长时间超过其额定功率使用导致的发热烧毁是发现得最多的	不要超限使用，它就是一个很长寿的器件
3	防火线路板	强度稍低，一般是单面板。板材较脆	
3	高阻值电阻（一般是大于 100 kΩ 的电阻）	故障率高于一般电阻，电阻越大越容易失效、工作温度越高越容易失效	尽量少使用高阻值电阻，注意电路板的散热
3	焊点	焊点本来不算是一个元件，但是它也能影响电子产品的品质。这里假设所有焊点在制造的时候，都是良品。但是长期使用后，由于反复的热胀冷缩，发热元件连接脚的焊点最容易出问题，其次是重量很重的元件的焊点，被经常性的冲击和振动，也会容易出问题	注意焊接工艺，设计的时候，这些故障高发的焊点，焊点面积要大一些，不仅强度高，也容易散热
3	激光头	一般是读光盘的灵敏度下降，激光头老化后，会出现某些品质好的光盘可以读出来、某些差劲的光盘就很难读。这就是激光头老化的前期症状。工作波长越短的激光头，就越容易受灰尘指纹之类的东西影响 不过我个人感觉，2004 年以后的产品，激光头损坏的就很少了，可能是半导体工艺进步所致	清洗激光头或者适当调节激光头上的可调电阻，以加大发射激光功率







续表 21.1

失效频率	器件的种类	容易失效的原因与后果	降低失效率的对策
3	软电线(直径较细的多芯电线)	一般的故障是折断,在焊点的根部折断。但其折断的概率要低于单芯的硬线	尽量减少折叠的次数和幅度
3	陶瓷电容	失效的,一般就是漏电	漏电了,只能更换
3	压电陶瓷发声片	由于这种器件质地很脆,所以在外力挤压后经常发生碎裂;还有在它上面电镀的导电电极容易脱落,也是常见的故障	
2	DIP 开关	整体故障率不高,但是我买过一个批次的,DIP 开关接触不良,整批退货。或者是因为大电流而损坏了器件	使用品质优良的器件,不要在 DIP 开关中走大电流
2	磁头	很坚硬的器件,有且仅有见过长期使用后被磨损的个体。早期失效的,不仅没见过,也没有听说过	
2	小信号电缆	大部分故障体现在电缆端点的连接器上,还有如果电缆经常折叠、收放也会导致电缆内部折断,这一点最好的例子是立体声耳机,它在插头的尾部和耳塞根部的那一段电缆都很容易在内部折断	用软线作电缆,就比较抗折叠和收放。不过最好是尽量减少折叠的次数和幅度
2	无刷电动机	寿命比有刷电动机好无数倍,但是在一些高速旋转的风扇上(比如 CPU 的散热风扇),经常看见其轴承出问题,导致噪声大增、耗电上升	选用品质好的电机,并且噪声增大后,在轴承处注油
2	有源晶体	这种晶体,还没见过跑不起来的。频率的稳定度也无可挑剔。但是见过几个有源晶体频率跑到标称频率的 1/3	选用品质好的器件
2	驻极体麦克风	长期使用有可能会导致麦克风的增益下降	
1	LED	一般是因为过电压和过电流而损坏	只要电压和电流正常,它就是一个很长寿的器件
1	粗电线	由于其粗大,抗各种机械力破坏的能力就很不错。但是见过因为过流而烧毁塑料皮的,不过概率很低	







续表 21.1

失效频率	器件的种类	容易失效的原因与后果	降低失效率的对策
1	带 Mask ROM 的半导体芯片	失效率和一般晶体管和半导体器件相当	不要过压、过流、超温度使用，一般不会发生问题
1	电感	只要不超过额定电流，基本不会出问题。超过额定电流容易导致磁性饱和。还有插件的电感，在漆包线和焊脚的连接处的故障率稍高一些	不要超过额定电流使用，保证线圈中的峰值电流小于其额定值
1	动圈式麦克风	我从没见过损坏的个体，听朋友说过在拆装过程中，不小心导致话筒的线圈折断	
1	环氧板	故障率很低的电路板，很少出故障。抗潮湿环境和强度都无可挑剔。如果有故障通常就是双面板过孔上下不导电	如果没有特别的必要，尽量不要用小过孔
1	金属膜电阻	故障率很低的器件，稍微超过一点额定功率，一般也不会损坏	只要不超过额定功率，几乎没有失效的
1	水泥电阻	水泥电阻一般都是大功率电阻。损坏的水泥电阻一般是超过其额定功率而损坏的	只要不超过额定功率，几乎没有失效的
1	碳膜电阻	故障率很低的器件。如上所述，高阻值电阻的故障率较高	只要不超过额定功率，几乎没有失效的
1	信号变压器	故障率不高，但是一些劣质的信号变压器，在漆包线和焊脚的连接处容易出问题，或者漆包线假焊、或者变压器支架断裂	
1	扬声器	常见的应该是线圈折断。我发现过我家的一个高端音箱上的低音扬声器中央塑料防尘膜老化开裂，不过还算运气好，耳朵听不出明显失真。低档的扬声器，比较容易出现音圈和磁体碰撞的故障，导致杂音	整体上说，它的失效率还是很低的
1	一般电阻	故障率很低的器件	只要不超过额定功率，几乎没有失效的

续表 21.1

失效频率	器件的种类	容易失效的原因与后果	降低失效率的对策
1	一般晶体管和半导体器件	损坏的概率很低，如果没有过流、过压和高温的话，一般失效概率低于 1/1000～1/10000	不要过压、过流、超温度使用，一般不会发生问题
0	涤纶电容	在电子行业混了十几年了，还真的没有见过一个失效的涤纶电容。书上说它被击穿后也能自恢复	这种电容好是好，就是太贵，而且体积较大
0	散热片	除了暴力冲击导致的损坏以外，还没有见过其他形式的损坏	

表 21.1 中的“失效频率”只是一个估值数字，是我从事设计工作十几年来对各种元件损坏概率的大致估值。最小为 0，最大为 10。数字越大说明越容易失效。

相对接近真实的失效概率可以这样计算：

现在以无刷电动机和有刷电动机二者的损坏概率作对比，看看无刷电动机的寿命是有刷电动机的几倍？

为便于讨论，我们约定：

- 由表 21.1 得知：无刷电动机的失效频率是 2，把它记为 b。
- 有刷电动机的失效频率是 6，把它记为 a。
- 比较接近真实的无刷电动机的失效频率，记为 B。
- 比较接近真实的有刷电动机的失效频率，记为 A。

计算结果显示：无刷电动机寿命是有刷电动机的寿命的 16 倍。

$$\frac{A}{B}=\frac{2^a}{2^b}=\frac{2^6}{2^2}=16$$

需要注意的是，这个结果仍然只是一个大致的评估值，只是更加接近真实的失效概率。

Retro Workshop 果粉工作室



附录

插柳不让春知道

技术人员有属于这个群体的可贵精神和优良品质，有在社会发展中不可或缺的地位和作用。他们是各行各业的骨干力量，能在关键的时候大显身手，创造了无数的奇迹，简直是无坚不摧。但是，我们也不讳言，他们这种工作上“硬作用”远较自身所起的“软作用”突出，他们当中有不少人都有一种似乎是不很被察觉又暗自留恋的处世哲学，即或多或少地有一种认为自己的技术学好了，技术过硬了就什么都不怕的狭隘心理。这就无形中影响了自身的发挥与发展。

21ic 网友大道至简写的《感悟设计：电子设计的经验与哲学》一书的消息一经公开就给人一种惊喜莫名，所以这样，也许关键之处在于它跳出了冰冷的纯技术范畴，将哲学思想与技术融为一体，将理解和技术融为一体。如果能将这些哲学思想以及理解技巧与学习提高和处世营生结合到一起，有意识地解决上述“软作用”不足和“狭隘心理”的问题，无疑能为人们将技术用于事业发展开辟了新的思路。虽然后来考虑此书初出，内容上和整体上还有一定的局限性，最后改名为《感悟设计：电子设计的经验与哲理》，但这并不妨碍该书的开创性，不妨碍它从一个侧面给技术人员提出一个现实的问题。这本书到底有多少价值，有多大意义？我想，关键在于我们对哲学的理解，搞清楚哲学到底与我们有什么关系，哲学离我们有多远，哲学是时刻伴随着我们还是根本就与我们无关？这是关于书籍方面的问题。

在读者方面，就像 21ic 电子论坛的网友 fushaobing 说的那样，国内的工程书籍大都就技术而讲技术，“没有作者自己的话，没有作者自己的思想，没有启发性”。估计提到这个问题，人们或多或少都有这样的感觉。这一方面正好说明了广大读者的需求，另一方面也正好说明这本集技术性、经验性和思想性的为一体的新书，会弥补现状的种种不足，多少会令冰冷的技术突现它有血有肉、有灵有魂的一面，多少会意味着“咱一线工程师里出类拔萃的人的涌现，国内工程书籍开始有点生机了。”（网友 Reflecter 语）。因此，从某种意义上说，技术人员如何

析出他们的思想，如何让他们的智慧更好地传播与交流，也许该书能给出一个有益的启发。

进一步地说，这本书显然是可以通过征集广大读者，尤其是网友的文章来加速完善和升级。我想，不管是基础的还是实用的，只要包含了技术性、经验性和思想性就是可选的好东西。我虽然没有真正搞过设计，但作为一种参与，在此通过一篇短文来回顾我与现代信息技术之本——二进制数的往事来表达对作者的敬意，来述说我的一点点感悟。

选择这个题材，一是因为我有一段特别的经历，二是因为二进制数是信息技术的基础，不管什么专业都有用。再说了，据我接触，别说是初学者，就是许多学有所成的人都觉得“十－二进制数的转换”有点烦，有时拿起笔来还不定一时不知道怎么做了。可在日常的学习、工作交流中，却又要时不时地碰到，更有意思的是，如果考试要考二进制数往往就是考“十－二进制数的转换”，是不是因为它是常见的几个数制间的转换中“难度”最大的一个呢？我不知道，但本文将趁此机会把这个“难度”给削下去。

1. 接触二进制数前的一段往事

让大家跟着我的一段往事进入我们的主题。

那是高考报到入学前的事了。那年的暑假，我到姨妈家帮忙插秧，不想姨父给我出了一道“难题”，后来我才知道它是和二进制数有密切联系的题。题目是这样的：

> 一片果林的成熟季节又到了，大批量的水果就要和买主交易，然而，买主来收购之前给农户出了一个“难题”：将 1 000 个水果事先放到 10 只箩筐里，待其到来时如果能整筐整筐地搬出来并凑够临时指定的数量，买卖照做还坚持长期合作；如果做不到，则预约好的买卖取消，今后也不来打交道了。

题目出得富有传奇色彩，很有魅力。姨妈也想一睹我的“风采”，年轻的我当时就信以为真般求解，但是，我当时就是给不出答案来。我呢，想着人家到来要求凑数一个两个，三个四个五个都是可能的，因此，什么可能都要考虑进去。可是一个两个都浪费掉一个箩筐，还有几百上千个水果呢。我当时是想到了第 1 筐 1 个，第 2 筐 2 个，之后每一筐都是已装筐水果的总和加 1 个水果，但是，不幸的是，当我看到很小的数量就占去很多箩筐的情况后，就一下子没了信心。你想，光 7 个水果就用掉了 3 个箩筐，装到 31 个水果的时候已经用掉一半的箩筐

了，哪还有信心算下去呢？

一个多月后，在学校学到了原来想都不敢想的“计算机”，那种光荣感就甭提了。入学一段时间后，我意外地发现奇妙的二进制数位权序列正好也是这样，即“低 n 位的位权之和再加上 1 总是等于较高一位的位权”，而且书上明讲了这是“最省资源”的。我这才好好地验证了一遍，结果发现我当初的想法是对的，接着立即给姨父去信说出我的答案：⑩(512－23＝)489，⑨256，⑧128，⑦64，⑥32，⑤16，④8，③4，②2，①1。但是，事情解决了也高兴不起来，毕竟事情解决得太晚了——如果真有那么一回事，我当初一时的怀疑和犹豫岂不是坏了大事了？我后悔当初没有增加一条思路：如果当时我认为不行，验算下去否定这个题目不就行了？可是我没有做，失去一时的“荣耀”，让姨妈失望是小事，若真误了事情不就是大事了?!

事情至此已告一段落，但是，也不得不说，这事和二进制数一样，这多年来都一直跟随着我。“相信自己”，说起来容易，但到实际就可能大打折扣，许多时候往往是“激流傻退”而不是一往无前。也许人生中最难也是最值得回味的事恐怕就是如何把握进退，特别是关键时刻偏偏被所谓的“经验”和“感觉”阻碍了人们的前进步伐的时候。

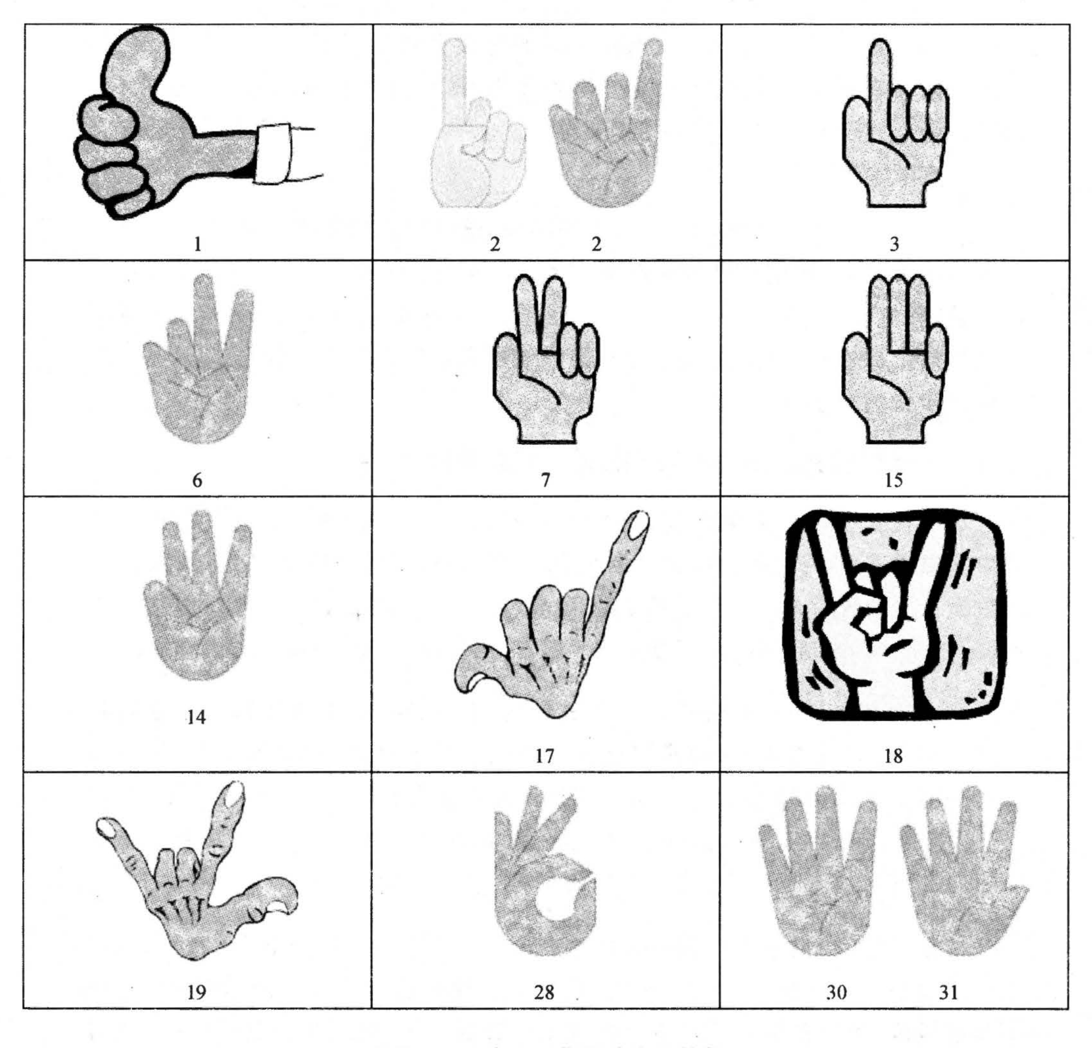
2. 一种应用二进制数的指算法(指操)

后来我居然因此事得到了意想不到的“补偿”。一边是因为我更爱思考，有事没事多想些道理，一边因为解决这个问题后，我后来找到一种简便实用的转换十进制数为二进制数的方法。实际情况是，我先“发明”了用手指算数数数的方法才想通那个“十－二进制数的转换”方法的。两者是有关联的，先说说这个指算法：

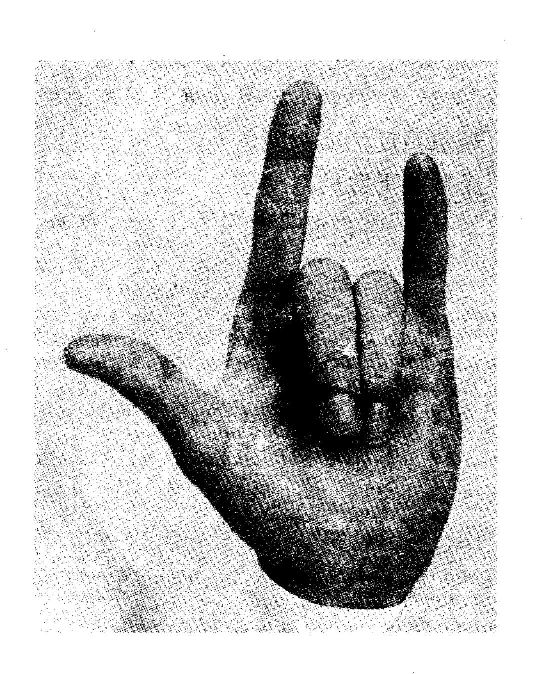
> 一只手掌 5 个手指，假设我们规定拇指、食指、中指、无名指和小指分别代表 1、2、4、8、16 这 5 个数(顺序倒过来或搅乱顺序也可以，规定好就行)，那么，在 0～31 以内的各个整数都可以通过手指的屈伸来表示了。例如要表示 25 这个数，因为 25＝16＋8＋1，所以，把小指、无名指和拇指伸出来，食指和中指不伸出去就 OK 了。再如，平常做的“OK 手势”表示的就是 28，如此等等。

清楚了吧，一只手居然能很清楚地表示 0～31 个数。扩大下去，一双手可以准确无误地表示高达 0～1023 的数，只是难度加大后有些人可能适应不了。为了降低难度，可以让两只手对应手指表示的数都一样，即用双手表示 0～62 范围内的数。

由于表示一个数的过程，有不少加减运算，故权且称之为一种“指算法”。从以往的经验来看，预计这种方法对小孩初期教育很有益，能让他们在不知道不觉中学会了一些二进制数的应用，将来学到二进制数的时候，一点就通。过去，我每次教小孩都说非常好玩，凡是教过的，没有不喜欢的，也许其中的魅力包括了难度不小各种手指屈伸动作。从这个意义上讲，又可以说这是一套很好的“指操”和“脑操”，对什么年龄段的人都很有价值。附图 1 和附图 2 是从图行天下网、昵图网和 e 库素材找来的几个指法示范。



附图 1　13 个用手指屈伸表示的数



附图 2　一个难度较大的手势/19
（原图来自 e 库素材）

3. 一种易学的“十一二进制数”快速转换方法

上述指算法（指操）其实也“隐含”了一种“十一二进制数”的转换方法。为了便于比较和弄清它的原理，下面让我们结合常用的“除 2 取余法”进行说明。

“除 2 取余法”的数学原理在众多的电子技术和计算机教材中多有介绍。如清华大学电子学教研组阎石主编的《数字电子技术基础（第 5 版）》所讲的那样，我见过的教材在介绍“十一二进制数”的转换方法时都这样，但这种方法在人工转换时往往觉得它并不自然、方便。其数学原理如下：（=0 或 1）

设十进制整数$(S)_{10}$等值于二进制数$(k_nk_{n-1}\cdots k_0)_2$，则有

$$(S)_{10}=k_n2^n+k_{n-1}2^{n-1}+\cdots+k_12^1+k_02^0=$$
$$2(k_n2^{n-1}+k_{n-1}2^{n-2}+\cdots+k_1)+k_0 \quad (1)$$

即对(1)式两边除以 2，所得的余数就是最低位k_0，($k_0=0$ 或 1)。将所得的商再进一步除以 2，所得的余数（0 或 1）就依次为$k_1,k_2,\cdots,k_n$，由此解出$(S)_{10}$的二进制数$(k_nk_{n-1}\cdots k_0)_2$。

这个转换方法在人工计算时，不少人对此很“感冒”。但是，我们知道，只要

是整数或可以完成精确转换的小数或带小数的数，不管是“十一二进制数”的转换还是“二一十进制数”的转换，两者肯定是有唯一对应关系的；如果是不能完成精确转换的，只不过是数值精度的取舍问题而已。既然如此，进行不同数制的数转换时，都是找出一个新数制下数的对应位的位权及其系数而已。对于二进制数而言，虽然各位的位权是不同的，但系数只有 1 和 0 两种。于是，我们可以有下面的计算方法：

观察(1)式的各项，即第 n 项，第 $n-1$ 项，…，第 1 项，第 0 项，可以发现，从左边开始，第 n 项的值(位权)必然小于或等于原来的十进制整数$(S)_{10}$。就是说，如果第 n 项与$(S)_{10}$相较，如果够减，说明后面的项还不一定全为 0，且本项的系数为 1；如果相减的结果为 0，则本项的系数为 1，且后面各项的系数均为 0；如果不够减，则该项的系数为 0，否则，所得的二进制数就必然大于原数$(S)_{10}$。由此可以判断 k_n到底是 1 还是 0，特殊情况下甚至可以一步求出结果。比如，有一个数 1 024，你可以直接知道它就是$(2^{10})_{10}=(10\ 000\ 000\ 000)_2$，如果这个数大于等于$(1024)_{10}$且小于$(2048)_{10}$，$k_{10}=1$ 是毫无疑问的，如果它小于$(1024)_{10}$，k_{10}必等于 0。归纳起来就是：

如果第 i 项与剩下的十进制数(也可以是原数)不够减，则有 $k_i=0(i=0,1,2,\cdots,n-1,n)$，对应项的值为 $k_i2^i=0$，而只有第 i 项才与剩下的十进制数(也可以是原数)够减，才有系数 $k_i=1(i=0,1,2,\cdots,n-1,n)$。当所有项的系数均为 1 时，各对应项的值(位权 2^i)如下式所示：

$$2^n+2^{n-1}+\cdots+2^i+\cdots+2^1+2^0=2^n+2^{n-1}+\cdots+2^i+\cdots+2048+1024+512+256+128+64+32+16+8+4+2+1 \quad (2)$$

既然既定项的位权是固定的，判断第 i 项的系数 k_i是 0 还是 1 也很容易，那么，只要每次按部就班地从够减的最大数开始，按顺序从大到小一直减下去就可以求解出各位的系数 k_i值了。现以原书的例子 173 为例，说明具体的解法：

173	本数为 173，256 不够减，128 够减，$k_7=1$，k_8及以上更高位都为 0。
173－128＝45	余数为 45，64 不够减，32 够减，$k_6=0$，$k_5=1$。
45－32＝13	余数为 13，16 不够减，8 够减，$k_4=0$，$k_3=1$。
13－8＝5	余数为 5，4 够减，$k_2=1$。
5－4＝1	余数为 1，2 不够减，1 够减，$k_1=0$，$k_0=1$。

求解结果为$(k_nk_{n-1}\cdots k_0)_2=(k_7k_6\cdots k_0)_2=(10101101)_2$

验证求解结果$(10101101)_2=(128+0+32+0+8+4+0+1)_{10}=(173)_{10}$，说明上述求解正确。

这种解法明显比原来的“除 2 取余法”快速得多，容易理解得多，但它并不深奥，它其实就是“二－十进制转换”的逆运算。对于带小数的情况，原理也相同。

一个十进制数转换成二进制数后，哪位该是 1 哪位该是 0，是完全固定的，结果肯定是唯一的，不同的只能是求解方法，想办法找出来就是了。一般情况下，求解步骤少，出错率低，效率高的方法才受欢迎。请读者验证本文介绍的方法是否具有这些特点，但应用时要注意，与“除 2 取余法”从低位开始，先求 k_0 的方法不同，本文介绍的方法是从高位开始，即先求 k_n 的。至于编程时是否比原来的“除 2 取余法”更方便，程序运行是否更快，我倒是没有试过，留给有兴趣的读者了。

4. 插柳不让春知道

我无意中得来的有关二进制数的“故事”至此讲完，希望读者从中有所裨益。在我看来，这些故事算得上是颇为不错的收获。与其说这是“无心插柳柳成荫”，不如说这是“插柳不让春知道”。当年姨父栽给我的难题，使得我在以后的学习和工作中更加不敢懈怠，更爱思考。

“插柳不让春知道”，据说是一句农谚，不太清楚其中的科学道理，但作为一种战术，在实际生活中却时常被人们运用，且收效往往非比寻常。营销就是其中的一个典型。“插柳不让春知道”在营销中的意思，就是在做广告、搞宣传之类的促销活动时，策划者会设法让你在不知不觉中了解他们平时想让你了解却又没有了解，特别是想让你相信平时又难以相信的信息，然后不露痕迹地牵走你的心，暗中达到他们秘不可宣的目的。比如，我国的“绿伞事件”就是个著名的案例——某面临困境的公司在一次活动中免费送伞三千给遇雨人，只要求事后五天内归还，但是时间到后却无一归还，结果，此事被媒体炒得热火朝天，由此产生的广告效应让老板走出困境，大发横财。当初的送伞行动，如愿以偿地换来了至高无上的广告策划效益。

这是一个获得巨大成功营销案例。据说，营销的最高境界就是“插柳不让春知道”。《感悟设计：电子设计的经验与哲理》一书从更深层次很有改造我们的学习，促进我们的交流的作用。如果说大道至简的新书是一次“插柳”，则它与营销

最怕别人知道"插柳"的目的不同,这是一次"插柳就让春知道"的"插柳"——我预料今后的技术交流深度和技术传播模式理应受到一定的影响,甚至有所转变,特别是在网络技术的发展,人们的交流更加方便、规模更大的大背景下。会不会是这样呢?让我们共同期待。

当然,在期待的同时我也想说"插柳最怕春知道"。只要春风一到,柳树就会生根,就会发芽,就会成长,直到杨柳依依……一切都会现其"原形"。有的事一旦被认同就很容易风行,但也须知一棵柳树成长起来不容易,因此,不能害怕"苦行僧"般的成长过程,那种希望"一插而就"的想法是不现实的。1939 年,毛泽东在《在延安在职干部教育动员大会上的讲话》中说到了"有期大学"和"无期大学"的概念,说"无期大学"是一个新发明,是一个新发明的大学制度。这个概念到了今天,意义还是非常重大的,但读这种"大学"也是一种"苦差",可是真正的人生却正是这样的人生,要想真正享受人生就得必须接受这种"苦差"。很多人以为读几年大学或多进修几个文凭就有了混世的本事,其实,我觉得这根本不是那么回事。我认为,读大学其实就像给自己的电脑安装了一个操作系统(插柳),没有别的应用程序,还是只能做一些简单的事情,因此,在参加工作后的"无期大学"不断进修才是最主要的,人的许多认识,人的许多能力、许多感悟,许多不完善的认识逐步完善甚至错误认识的修正,特别是关键能力就是在这一时期得来的。这个时期的作为,才是真正决定你在人生的春天到来的时候,自己的操作系统到底安装了哪些应用程序,能干什么事,达到什么档次(水平)。我想,这才是"天之骄子"的正常理解。

黄中宁
2009 - 04 - 21



此文由热心人士黄中宁(网名 iC921)提供,在此表示特别感谢。

2009.2.20



策划编辑：胡晓柏
封面设计：runsign 瑞正设计

感悟设计
电子设计的经验与哲理

内容特色

作者从事电子类的工作十几年了，其中纯粹设计工作也超过了10年。每天都和软件和硬件打交道。总结这些年来，学到了一点知识，得到了一点经验，也感悟了一点东西。

本书主要从设计者的思维的角度来展开叙述。但和别的讲设计的书籍显著的不同在于：本书不讲某个专题，而是注重描述设计的整个过程的思维方式和解决问题的方法。

本书作为一本讲述电子设计的书籍，却在其中大量穿插了很多非电子的案例和知识点。因为无论什么学科的知识，都是用来解决问题的。既然是用来的解决问题的，那就会有很多共性，因为即使是不同学科的问题，它们本身也是经常可以归纳出共同点的。所以把这些东西贯穿起来，更容易感悟出事物的内在规律。同时，设计工作作为一项思想性很强的工作，自然离不开一些哲学的规律。所以，书中也引用了一些大家都耳熟能详的哲学道理，并把这些道理作为文章的标题，用案例来陈述这些哲学道理的合理性。这些哲学道理可能就是我们所说的“只可意会，不可言传”的东西。而本书就是致力于写出这些“只可意会，不可言传”的东西。

读者对象

电子设计开发人员、管理人员、培训人员，以及电子设计方向研究生、本科生。

上架建议：单片机/嵌入式系统

ISBN 978-7-81124-573-8



定价：32.00元